HAL
open science

Cloud-Radio Access Networks : design, optimization and algorithms

Niezi Mharsi

To cite this version:

Niezi Mharsi. Cloud-Radio Access Networks : design, optimization and algorithms. Networking and Internet Architecture [cs.NI]. Université Paris Saclay (COmUE), 2019. English. NNT : 2019SACLT043 . tel-02373002

HAL Id: tel-02373002

https://pastel.hal.science/tel-02373002

Submitted on 20 Nov 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

NNT : 2019SACLT043

Cloud-Radio Access Networks : Design, Optimization and Algorithms

Thèse de doctorat de l'Université Paris-Saclay
préparée à Institut Mines-Télécom - Télécom ParisTech

Ecole doctorale n°580 Sciences et Technologies de l'Information et de la Communication (STIC)
Spécialité de doctorat : Réseaux, Information, Communications

Thèse présentée et soutenue à Paris, le 10 Octobre 2019, par

NIEZI MHARSI

Composition du Jury :

Djamal ZEGHLACHE Professeur, Institut Mines-Télécom - Télécom SudParis	Président
Adlen KSENTINI Professeur, Eurecom	Rapporteur
Lina MROUEH Maître de conférences - HDR, Institut Supérieur d'Electronique de Paris	Rapporteur
Loutfi NUAYMI Professeur, Institut Mines-Télécom - IMT Atlantique	Examinateur
Philippe MARTINS Professeur, Institut Mines-Télécom - Télécom ParisTech	Directeur de thèse
Makhlouf HADJI Chercheur - HDR, Institut de Recherche Technologique - IRT SystemX	Co-directeur de thèse

Thèse de doctorat

Contents

List of Figures **iii**

List of Tables **vi**

List of Acronyms **ix**

Introduction **1**

1 Context, motivations and contributions **3**

1.1 Context and motivations : Cloud-Radio Access Network (C-RAN) . . 3

1.1.1 C-RAN architecture 4

1.1.2 C-RAN benefits 6

1.1.3 C-RAN challenges 7

1.2 Contributions 8

1.2.1 Low-complexity algorithms for constrained resource allocation problem in C-RAN 9

1.2.2 Mathematical programming approach for full network coverage optimization in C-RAN 10

1.2.3 Cost-efficient and scalable algorithms for BBU function split placement in C-RAN 11

1.3 Publications 11

2 C-RAN optimization: mathematical background and state-of-the-art **13**

2.1 Introduction 13

2.2 Mathematical background 13

2.2.1 Combinatorial optimization : techniques and algorithms . . . 13

2.2.2 Simplicial homology background 23

2.3 C-RAN optimization : state-of-the-art 25

2.3.1 Constrained resource allocation problem 25

2.3.2 Network coverage optimization problem 27

2.3.3 BBU functions split and placement problem 29

2.4 Conclusion 32

3 Constrained resource allocation in C-RAN **33**
3.1 Introduction . . . 33
3.2 Problem statement . . . 34
3.2.1 System model . . . 34
3.2.2 Problem complexity . . . 37
3.3 Proposed algorithms . . . 37
3.3.1 Integer linear programming formulation . . . 38
3.3.2 Matroid-based approach . . . 40
3.3.3 b-Matching formulation . . . 42
3.3.4 Multiple knapsack-based approach . . . 44
3.4 Performance evaluation . . . 45
3.4.1 Simulation settings and parameters . . . 46
3.4.2 Performance metrics . . . 47
3.4.3 Performance analysis . . . 48
3.5 Conclusion . . . 55

4 Full network coverage optimization in C-RAN **57**
4.1 Introduction . . . 57
4.2 Problem statement . . . 58
4.2.1 System model and problem description . . . 58
4.2.2 Problem complexity . . . 60
4.3 Branch-and-Cut formulation . . . 61
4.3.1 Convex hull characterization . . . 61
4.3.2 New valid inequalities . . . 64
4.3.3 Complete mathematical formulation . . . 67
4.4 Performance evaluation . . . 69
4.4.1 Simulation parameters and settings . . . 69
4.4.2 Performance metrics . . . 69
4.4.3 Simulation results and performance analysis . . . 70
4.5 Conclusion . . . 74

5 BBU function split placement in C-RAN **75**
5.1 Introduction . . . 75
5.2 Problem statement . . . 76
5.2.1 BBU function split modeling . . . 76
5.2.2 Network topology description . . . 77
5.2.3 System model . . . 78
5.2.4 Problem complexity . . . 79
5.3 Exact mathematical approach . . . 80
5.4 Approximation approaches : multi-stage graph algorithms . . . 84
5.5 Performance evaluation . . . 86
5.5.1 Simulation parameters and settings . . . 86
5.5.2 Performance metrics . . . 87
5.5.3 Simulation results and performance analysis . . . 87
5.6 Conclusion . . . 91

6 Conclusions and perspectives **93**

Bibliography **95**

List of Figures

1.1 5G services and opportunities. Source: Cisco, 2018 [1] 3
1.2 Traditional RAN to C-RAN architecture 5
1.3 C-RAN architecture and components 6

2.1 An example of a convex polytope 15
2.2 An example of linear inequality separating the convex polytope P and the solution x^* 17
2.3 Example of k-simplices 23
2.4 Example of simplicial complex 24
2.5 Delaunay triangulation. Source: [2] 28
2.6 3GPP functional split options 30
2.7 Overview of considered functional split options 31

3.1 System model for constrained resource allocation problem 34
3.2 A solution example of the constrained resource allocation problem . . 35
3.3 Example of simulation scenarios for RRH-BBU assignment problem . 46
3.4 Matroid-based approach : rejection rate variation when increasing number of edge data centers 50
3.5 SLA violations rate behavior of the matroid-based approach 51
3.6 Resource utilization in different space dimensions 52
3.7 Real trace : Orange 4G-LTE cell map in Paris. Source: [3] 53

4.1 System model: graph construction based on antennas positions and interference 58
4.2 A graph solution example of the full coverage network problem 60
4.3 Each solid line edge (i, j) is necessary in the final graph/solution . . . 62
4.4 Example of edge intersection (interference) 62
4.5 Example of graph solution containing a hole $(v_2, v_4, v_5, v_6, v_2)$ 64
4.6 Example of a chordless cycle $(v_2, v_3, v_5, v_8, v_2)$ of size 4 65
4.7 Example of two connected components (triangulations) creating holes in the final graph 66
4.8 An Orange 4G-LTE cell map: **before Branch-and-Cut optimization** 73
4.9 An Orange 4G-LTE cell map: **after Branch-and-Cut optimization** 73

5.1 BBU function split modeling for each antenna demand 77
5.2 Physical network architecture 77
5.3 System model for BBU function split placement 78

5.4 Example of Virtual Network Embedding problem. Source: [4] 80
5.5 A multi-stage graph example 84
5.6 Algorithms' convergence time using 20 edge cloud data centers 87
5.7 CPU residual resources behavior 89
5.8 Latency behavior 90

List of Tables

3.1 RRH-BBU assignment problem : variables and parameters 36
3.2 RRH-BBU assignment algorithms : simulation settings and parameters 47
3.3 Performance of the exact approach based on ILP formulation 49
3.4 Heuristic algorithms' performance assessment 49
3.5 Performance evaluation using a real cellular network in Paris 53
3.6 Algorithms' scalability assessment 54
3.7 Algorithms' qualitative comparison 54

4.1 Network coverage optimization : simulation settings and parameters . 69
4.2 Exact algorithm performance: convergence time to the optimum . . . 70
4.3 Performance comparison : ILP vs Rips approach 71

5.1 BBU function split placement problem : variables and parameters . . 79
5.2 Algorithms' performance comparison : ILP vs heuristic variants . . . 88
5.3 Scalability and convergence time comparison using Euclidean graphs . 90

List of Acronyms

- **BBU**: BaseBand Unit
- **BFD** Best Fit Decreasing
- **CAPEX**: CAPital EXpenses
- **COMP**: Coordinated Multi-Point
- **CPU**: Central Processing Unit
- **C-RAN**: Cloud-Radio Access Network
- **C-RoFN**: Cloud-Based Radio over Optical Fiber Network
- **DSP**: Digital Signal Processing
- **eMBB**: enhanced Mobile BroadBand
- **eNodeB**: evolved Node B
- **EPC**: Evolved Packet Core
- **FFD**: First-Fit Decreasing
- **FPGA**: Field-Programmable Gate Array
- **HARQ**: Hybrid Automatic Repeat reQuest
- **ICIC**: Inter-cell Interference Coordination
- **ILP**: Integer Linear Programming
- **IoT**: Internet of Thing
- **LP**: Linear Programming
- **MAC**: Medium Access Control
- **MILP**: Mixed Integer Linear Program
- **mMTC**: massive Machine-Type Communications
- **mmWave**: millimeter Wave
- **MWT**: Minimum Weight Triangulation
- **NGMN**: Next Generation Mobile Networks
- **NG-PoP**: Next Generation Point of Presence

- **OAI**: OpenAirInterface
- **OPEX**: OPerating EXpenses
- **PDCP**: Packet Data Convergence Protocol
- **PHY**: Physical layer
- **QoS**: Quality of Service
- **RAN**: Radio Access Network
- **RF**: Radio Frequency
- **RLC**: Radio Layer Control
- **RRH**: Remote Radio Heads
- **SA**: Simulated Annealing
- **SDN**: Software Defined Network
- **SDR**: Software Defined Radio
- **SLA**:Service-Level Agreement
- **TCO**: Total Cost of Ownership
- **UE**: User Equipment
- **URLLC**: Ultra-Reliable and Low-Latency Communications
- **VM**: Virtual Machine
- **VNE**: Virtual Network Embedding
- **VNR** Virtual Network Requests
- **V2X**: Vehicle to anything
- **WSN**: Wireless Sensor Network
- **3GPP**: Third Generation Partnership Project
- **4G**: Fourth Generation
- **4G-LTE**: Fourth Generation-Long Term Evolution
- **5G**: Fifth Generation

Introduction

The main goal of this thesis is to investigate the deployment of Cloud Radio Access Network (C-RAN) and its relevant and exciting research challenges such as resource allocation and placement problems, network coverage optimization, etc. In fact, with the exponential growth in data traffic demands, C-RAN is seen as a key enabler for the next generation of mobile networks (5G) to handle the diverse service requirements and reduce network costs, including CAPEX (CAPital EXpenses) and OPEX (OPerating EXpenses). This architecture consists in decoupling the BaseBand Units (BBU) from the Remote Radio Head (RRH) and centralizing the baseband processing into common data centers (a pool of BBUs) offering resource utilization gains and cost savings.

In this thesis, we use combinatorial optimization techniques to propose new approaches that rapidly reach optimal or near optimal solutions for resource allocation problems in the context of C-RAN when guaranteeing good Quality-of-Service (QoS) for end-users' demands.

Chapter 1 provides the main motivations of our research works. The context of this thesis is then described by providing a brief introduction of C-RAN architecture, its benefits and challenges. We also provide an overview of our novel contributions and the list of scientific publications that we propose during the thesis.

Chapter 2 provides an overview of mathematical backgrounds used in this manuscript, including two major domains : (i) combinatorial optimization and (ii) simplicial homology and gives a survey on the most important challenges that we address in this manuscript to enable the deployment of C-RAN architecture.

Chapter 3 discusses an exciting research challenge in the context of C-RAN which is the problem of RRH-BBU assignment that aims to allocate limited computing resources in common edge data centers to the heterogeneous antennas demands when meeting strong latency requirements. We propose a complete mathematical formulation based on Integer Linear Program (ILP) to describe the convex hull of the RRH-BBU assignment problem and provide optimal solutions. For sake of scalability, we introduce new approximation algorithms to rapidly find good strategies to assign antennas demands to centralized data centers, under latency and processing requirements. A performance evaluation of our proposed algorithms is discussed using different simulation scenarios and performance metrics.

Chapter 4 presents an exact approach based on Branch-and-Cut methods to reduce inter-cell interference, caused by the high density of cells in C-RAN, when guaranteeing a full network coverage. We use simplicial homology-based approaches to evaluate the ability of our approach in finding optimal solutions in acceptable times.

Chapter 5 focuses on another major challenge to enable the deployment of C-RAN which consists in finding optimal trade-offs between benefits of processing centralization and strong latency requirements. In this chapter, we propose exact and heuristic algorithms based on combinatorial optimization methods to determine

the optimal placement of baseband processing between RRHs and BBUs. The BBU functions placement is based on different split configurations and transport network characteristics. Two simulation scenarios and different performance metrics are proposed to benchmark our proposed algorithms.

Finally, Chapter 6 concludes the manuscript and investigates future research challenges.

Chapter 1

Context, motivations and contributions

1.1 Context and motivations : Cloud-Radio Access Network (C-RAN)

Mobile data traffic demands are exponentially increasing due to the rapid growth in the number of connected terminals and mobile devices (smartphones, tablets, etc). Furthermore, the promise of 5G networks is not only a simple evolution of 4G networks with higher peak throughput and larger spectrum bands, but also to deal with new services and new business opportunities that are classified by 3GPP [5] into three main families of use cases : (i) enhanced Mobile BroadBand (eMBB), (ii) Ultra-Reliable and Low-Latency Communications (URLLC) and (iii) massive Machine-Type Communications (mMTC).

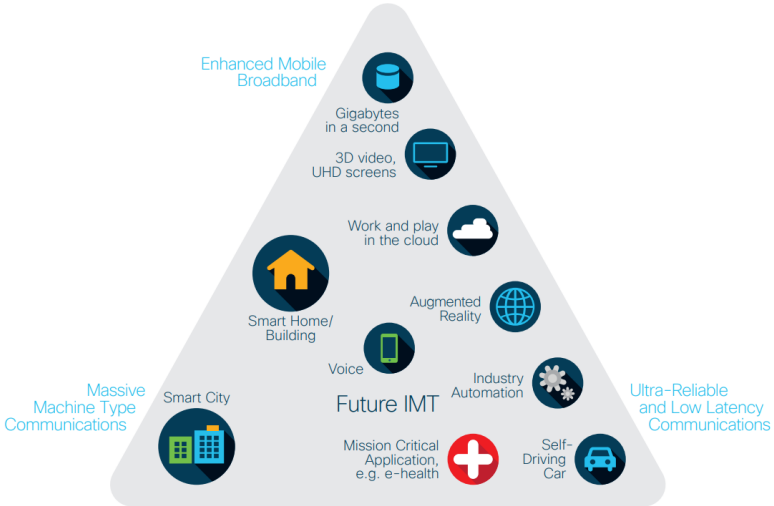


Figure 1.1: 5G services and opportunities. Source: Cisco, 2018 [1]

As depicted in Figure 1.1, eMBB services gather the set of use cases requiring high data rates across a wide coverage area, URLLC involve the applications that

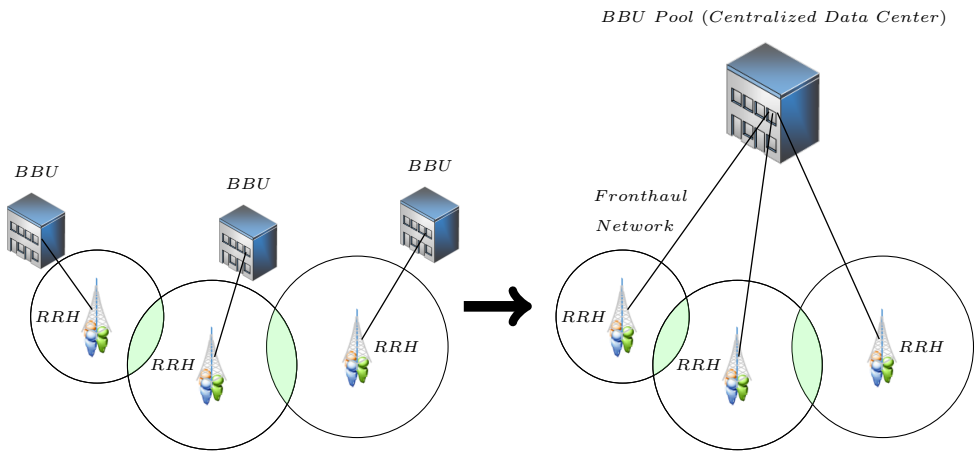
have strict requirements on latency and reliability such as Vehicle to anything (V2X) and industrial control applications and mMTC represent the need to support a very large number of devices in small area forming an Internet of Things (IoT).

The deployment of such technologies is significantly increasing the network costs, including CAPEX and OPEX, the amount of baseband processing as well as levels of inter-cell interference because of the high density of cells. However, current network architectures will no longer be able to deal with the demands for high data rates and lower latencies and to support new 5G services. Therefore, network operators need to investigate new network architecture for next generation of mobile networks to meet these challenges and reduce CAPEX and OPEX. In this context, C-RAN has been proposed as a promising network architecture to handle the diverse service requirements and guarantee a good QoS for end-users. Unlike conventional mobile networks where the baseband functions reside on the cell sites along with the antennas, C-RAN decouples the traditional base station into RRHs and centralized BBUs which will be pooled and used as shared resources, offering network resource utilization gains and energy efficiency. C-RAN is expected to minimize network costs (CAPEX and OPEX) by reducing the number of base stations needed to meet antennas demands. Meanwhile, C-RAN will improve radio performance by facilitating various forms of multi-cell coordination to handle the inter-cell interference, caused by the high density of cells.

In the following, we detail the fundamental aspects of C-RAN architecture and its main components. Then, we discuss the main advantages of this architecture and finally, we outline the most important challenges that we are facing while deploying C-RAN architecture.

1.1.1 C-RAN architecture

Current generation mobile networks are using traditional Radio Access Network (RAN) that consists in locating the radio and baseband processing functionalities in the same base station. In fact, as depicted in Figure 1.2, the traditional base station consists of two components, the antenna (RRH) and the BBU (data center), co-located at the same macro site (eNodeB). Nevertheless, these networks are no longer able to provide high data rates, meet strong latency requirements and guarantee high QoS for end-users' demands. In order to achieve these goals, C-RAN has been proposed as a promising architecture for next generation of mobile networks (5G) to handle the diverse service requirements. The main concept of C-RAN consists in decoupling the BBUs from the antennas and pooling the computing resources into centralized data centers, i.e. BBU pools. The BBU computation pools will be shared among multiple base stations in order to achieve resource utilization gains and network cost savings. Figure 1.2 illustrates the main difference between the traditional RAN (the left part of Figure 1.2) and C-RAN architecture (the right part of Figure 1.2) in terms of resource pooling and BBU centralization.



(a) Traditional RAN (b) C-RAN architecture

Figure 1.2: Traditional RAN to C-RAN architecture

As mentioned above, the proposed C-RAN architecture consists in centralizing the baseband processing of antennas demands from several cell sites in BBU pools when the RRHs are connected to the centralized data centers through fronthaul network. Figure 1.3 illustrates C-RAN architecture focusing on three main components:

- **RRHs or antennas**: are located at the cell sites and they forward the baseband signals received from User Equipments (UEs) to the BBU pools for centralized processing in the uplink while transmitting Radio Frequency (RF) signals to UEs in the downlink. RRHs perform radio functions including RF conversion, amplification, filtering, analog-to-digital conversion and digital-to-analog conversion [6].

- **BBU pool or centralized data center**: is a centralized location of computing and processing resources shared among multiple cell sites. In fact, each BBU pool can serve 10 to 1000 RRHs [7] and it consists in centralizing the processing of baseband signals of antennas demands and then optimizing the allocation of computing resources.

- **Fronthaul network**: is a set of communication links between RRHs and BBU pool. The fronthaul traffic exchanged between antennas and centralized data centers can be transmitted using typical protocols including OBSAI [8] and CPRI [9], which is the most widely used in cellular networks. Fronthaul network can be released by different technologies such as optical fiber communications, standard wireless communications, or millimeter Wave (mmWave) communications [10].

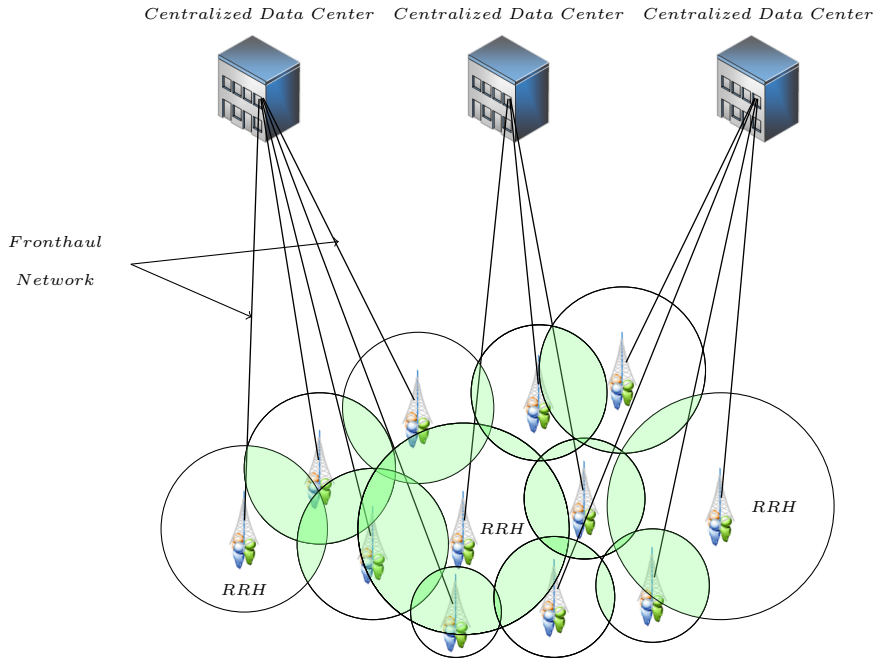


Figure 1.3: C-RAN architecture and components

1.1.2 C-RAN benefits

C-RAN architecture comes with many advantages (see for instance [11], [12], [13] and [14]) that will be detailed in the following :

- **Cost savings in CAPEX and OPEX**: centralization of computational resources in C-RAN enables an efficient utilization of BBU resources by reducing the total number of BBUs needed to meet end-user demands. This leads to achieve lower energy consumption and potential cost reductions in CAPEX and OPEX. In [11], authors conclude that C-RAN reduces CAPEX by 15% and OPEX by 50% when compared to a traditional mobile network.

- **Capacity and coverage improvement**: in C-RAN architecture, centralized data centers (BBU pools) are expected to host more end-user demands coming from different cell sites. In [7], authors claim that each BBU pool should support 10 to 1000 base station sites which enables to improve the network capacity by covering a larger area and serving more users than traditional base stations when guaranteeing high QoS.

- **Resource utilization gains**: since in C-RAN baseband processing of multiple cells is carried out in centralized BBU pools, resource sharing becomes feasible and hence the resource allocation can be more flexible and on demand unlike traditional networks. This enables to improve the efficiency of network resource utilization.

- **Inter-cell interference reduction**: in C-RAN, the centralized processing enables easy implementation of joint processing and scheduling algorithms, which

help to reduce inter-cell interference and improve spectral efficiency. Indeed, efficient interference management techniques such as Coordinated Multi-Point (CoMP) [15] and Inter-cell Interference Coordination (ICIC) [16] can be easily implemented in the BBU pool which help to optimize the transmission from many cells to multiple BBUs.

- **Network flexibility and extensibility**: With the centralization of computing resources in a common location, network operators will quickly deploy new antennas (RRHs) and connect them to the BBU pool to cover more service areas and make upgrades to their network infrastructures. This will improve the network scalability and flexibility and will facilitate the network maintenance.

1.1.3 C-RAN challenges

Despite all the benefits (as detailed above) brought by C-RAN architecture, a number of challenges need to be addressed to enable and facilitate the deployment of C-RAN. In this section, we focus on the main challenges of C-RAN that will be addressed in this manuscript.

- **Strong latency expectations on fronthaul network** : in C-RAN architecture, the fronthaul network that connects RRHs to BBU pools must carry a significant amount of data traffic demands in real time with high bandwidth and strong latency requirements. In fact, according to [11] and [17], the transmission delay on a link between RRHs and the centralized data centers should be kept below **1 millisecond** to meet HARQ[1] requirements which impose that the maximum distance between RRH and BBU pool must not exceed 20 to 40 kilometers [11].

  The fronthaul capacity and delay constraints can be alleviated by flexibly splitting the baseband processing between BBU pools and RRHs. The envisioned solution consists in moving a part of baseband processing functions in centralized data centers in order to reduce the increasing data throughput and the overheads of the signal transmitted through fronthaul network. The evaluation of different options of BBU functions split in C-RAN will be detailed in Section 2.3.3 of Chapter 2.

- **New resource allocation algorithms**: another key challenge of C-RAN is the assignment of computing and radio resources shared among multiple cell sites. Network operators should investigate new approaches to determine the best strategies to assign heterogeneous antennas demands to the available edge data centers when satisfying hard latency requirements. An optimal assignment between RRHs and BBUs enables to achieve resource utilization gains by reducing the number of edge data centers used to satisfy antennas demands. This will also minimize the network costs including CAPEX and OPEX. This exciting research challenge will be addressed in Chapter 3.

[1]HARQ (Hybrid Automatic Repeat reQuest) is the process that poses the most stringent delay requirement for cellular networks

- **Inter-cell interference reduction**: in C-RAN architecture, each BBU pool should be able to support 10 to 1000 RRHs [7] and provide optimal and joint baseband processing of antennas demands. Moreover, network operators need to increase the density of cells by deploying more antennas in order to meet the growing number of end-users' demands. This requires to investigate new algorithms to reduce inter-cell interference while guaranteeing the full network coverage. We will address this challenge in Chapter 4.

- **BBU functions placement**: the centralization of RAN functions in BBU pools allows to capitalize on computational gain and improve mobile network functions, e.g. scheduling and flow control. Nevertheless, the processing of RAN functionalities in a centralized location increases the fronthaul data rate demands and requires strict latency requirements. Therefore, there is a need for addressing the trade-off between RAN functions centralization and transport requirements by finding the optimal placement of baseband functions in C-RAN architecture. We will investigate in Chapter 5 new algorithms to determine the optimal locations of BBU functions under different transport requirements.

- **Virtualization**: In addition to the centralization of baseband processing, network virtualization is an important technique for the realization of C-RAN architecture. In fact, virtualization decouples the computing resources from the physical hardware and creates new virtual entities called Virtual Machines (VMs) which are responsible for handling BBU functions in each centralized data center (for more details, see [18] and [19]). Recent efforts investigated new software solutions to implement virtualized RAN. In fact, authors in [20] and [21] propose a new software platform called OpenAirInterface (OAI), which is an open-source Software Defined Radio (SDR) implementation for 4G/5G networks including both RAN and Evolved Packet Core (EPC) functionalities. This challenge has been very well studied in the literature (see for instance [22], [23] and [24]) and this is not in the scope of our work.

1.2 Contributions

This section highlights the main contributions of this thesis. Our contributions are divided into three parts : in the first part, we propose new resource allocation algorithms to efficiently assign antennas demands to centralized data centers under strong latency expectations and limited computing resources' constraints. The optimal assignment will reduce the fronthaul throughput and maximize the computational gain. Such gains are strongly constrained by reducing the inter-cell interference levels that increase because of high density of cells. Thus, in second part, we investigative an exact mathematical formulation to minimize inter-cell interference when maintaining a full network coverage. In the third part, we focus on determining optimal locations of baseband functions based on different split configurations and transport network characteristics.

1.2.1 Low-complexity algorithms for constrained resource allocation problem in C-RAN

C-RAN is a novel mobile network architecture which consists in centralizing the baseband processing in common edge data centers (sometimes referred to as Next Generation-Point of Presence (NG-PoP)) and then sharing the computing resources among different antennas (RRHs). This enables network operators to achieve efficient network utilization and cost savings. However, such gains can only be achieved by finding best strategies in the assignment of the edge data centers to the heterogeneous antennas demands while jointly reducing the resource utilization and the communication latency on the fronthaul network between RRHs and edge data centers. In this contribution, we propose a complete mathematical model based on Integer Linear Programming (ILP) formulation to identify the most appropriate strategies concerning antennas demands assignment to the available edge data centers. The proposed ILP formulation optimizes jointly the latency (transmission delay) on the fronthaul network and the resource consumption (expressed in terms of active edge data centers). Our proposed optimization model provides optimal solutions for small and medium problem instances. Thus, for larger problem instances, we propose three approximation algorithms, based on exact theories and approaches, that scale well, converge reasonably fast and provide good strategies of RRH-BBU assignment :

- **Matroid-based approach** : we propose a new approximation algorithm based on Matroid theory [25]. This approach models the constrained resource allocation problem as a graphic matroid representation to find good strategies to assign the edge data centers to the antennas demands. In fact, the objective of this approach is to achieve an optimal assignment, when jointly minimizing communication latency and resource consumption.

- **b-Matching algorithm** : we investigate an algorithm based on b-matching approach [26] that aims to find the minimum weight matching between antennas and edge data centers, with limited capacity of processing (CPU cores), when satisfying the expected communication latency. The proposed b-matching algorithm can reach optimal or near-optimal solutions for large network sizes.

- **Multiple knapsack-based approach** : another approximation algorithm will be proposed using multiple knapsack formulation which is very used in the literature to solve many variants of resource allocation problem (for instance [27], [28], [29] and [30]). Hence, the proposed multiple knapsack formulation will be used to evaluate the performance of the above heuristic algorithms.

A part of this work, in which we proposed an ILP formulation and matroid algorithm, has been published in IEEE International Conference on Smart Communications in Network Technologies 2018 (SaCoNet 2018) [31]. This publication includes also a performance assessment of the proposed algorithms using different simulation scenarios to quantify the scalability and the potential benefits of the discussed approaches in the context of C-RAN. Next to that, this work has been

extended by proposing two other approximation approaches, b-matching formulation and multiple knapsack-based algorithm, and a deep analysis to evaluate the performance of the proposed algorithms in terms of efficiency, scalability and ability to find good solutions in acceptable times using simulations and a real 4G-LTE network map. This part of work has been submitted in the International Journal of Computer and Telecommunications Networking (Computer Networks) [32].

1.2.2 Mathematical programming approach for full network coverage optimization in C-RAN

In C-RAN, network operators will increase the density of existing cells by deploying more antennas in order to enhance the network capacity and coverage and enlarge the network spectrum. However, cells densification comes with an increasing of inter-cell interference which causes serious degradations of the provided networks' QoS. Hence, network operators need new approaches to reduce inter-cell interference and maintain a full network coverage jointly. Our contribution consists in proposing a Branch-and-Cut algorithm to reach a good tradeoff between interference elimination/reduction and network coverage optimization in the context of C-RAN. In fact, we propose a mathematical description modeling the problem according to the RIPS approach based on simplicial homology [33] (more details can be found later in Chapter 2). Our mathematical model describes the convex hull of the discussed problem and allows to reach optimal solutions even for large problem instances. This description is then enlarged by new valid inequalities and cutting planes to better precise the polytope containing the optimal solution. This contribution based on polyhedral approaches optimization is new and has never been addressed in the literature to cope with the full coverage hole problem in C-RAN. To reach the above objectives, our contribution is described as follows :

- Minimize the number of coverage holes in the cellular network.
- Reduce or eliminate the inter-cell interference by adjusting the coverage radius of antennas without creating coverage holes in the final network.
- Rapid (polynomial time) detection of coverage holes.

In addition, we provide a deep analysis of the performance of our Branch-and-Cut algorithm using different simulation scenarios and a real network map to confront our algorithm to different infrastructures and network topologies. This allows us to evaluate the efficiency and reliability of our approach and the quality of the found solutions through different performance evaluations and metrics.

This work has been published first in the IEEE Global Communications Conference, GLOBECOM 2018 [34] and has been extended then to the International Journal of Computer and Telecommunications Networking (Computer Networks) [35].

1.2.3 Cost-efficient and scalable algorithms for BBU function split placement in C-RAN

As it was already mentioned, the main functionality of C-RAN consists in centralizing the baseband processing from multiple base stations into BBU pools (common data centers). This enables network operators to achieve many benefits in terms of cost savings and capacity increasing. However, the deployment of C-RAN requires very high capacity and low latency on the fronthaul links to connect antennas (RRH) to the centralized data centers (BBUs). Thus, the challenge is to find an optimal split of baseband processing between BBUs and RRHs in order to reach good tradeoff between benefits of centralization and high transport requirements. In this context, various functional splits have been proposed, each of which imposes different throughput and delay requirements. This contribution considers 3GPP RAN split [5] that is outlined as the best option in the 3GPP and it consists in splitting the baseband functions into three components : i) PHY layer ii) MAC and RLC layers and iii) PDCP layer. Accordingly and based on this split configuration, we aim to propose new optimization algorithms to determine optimal locations of baseband functions when considering strict transport requirements on the fronthaul network in terms of latency.

We propose an exact approach based on ILP formulation to optimally deploy BBU functions from multiple cells on the centralized data centers while jointly minimizing the network resource consumption and the end-to-end latency. The exact approach provides the optimal solution for small and medium problem sizes. For larger problem instances, we propose new heuristic algorithms based on the construction of an extended multi-stage graph to rapidly determine the optimal placement of the baseband processing functions when jointly meeting their CPU and latency requirements. These algorithms are benchmarked using different simulation scenarios to evaluate the efficiency and scalability of our algorithms as well as their ability to achieve optimal solutions in acceptable times.

This contribution has been published in IEEE Wireless Communications and Networking Conference (WCNC 2018) [36].

1.3 Publications

The scientific publications during this thesis are summarized in the following:

- **Journals**
  - N. Mharsi, M. Hadji, *A mathematical programming approach for full coverage hole optimization in Cloud Radio Access Networks,* **Computer Networks**, Volume 150, 2019, Pages 117 − 126, ISSN 1389 − 1286, https://doi.org/10.1016/j.comnet.2018.12.015. (http://www.sciencedirect.com/science/article/pii/S1389128618307928) (Chapter 4)

- N. Mharsi, M. Hadji, *Edge computing optimization for efficient RRH-BBU assignment in Cloud Radio Access Networks,* **Computer Networks**, submitted. (Chapter 3)

- **International conferences**

  - N. Mharsi, M. Hadji, P. Martins *Full Coverage Hole Optimization in Cloud Radio Access Networks,* **Globecom 2018 : IEEE Global Communications Conference,** December 2018 Abu Dhabi, UAE. (Chapter 4)

  - N. Mharsi, M. Hadji, *Joint Optimization of Communication Latency and Resource Allocation in Cloud Radio Access Networks,* **SacoNeT 2018 : IEEE International Conference on Smart Communications in Network Technologies,** October 2018 Algeria. (Chapter 3)

  - N. Mharsi, M. Hadji, D. Niyato, W. Diego, R. Krishnaswamy, *Scalable and Cost-Efficient Algorithms for Baseband Unit (BBU) Function Split Placement,* **WCNC 2018 : IEEE Wireless Communications and Networking Conference,** April 2018 Barcelona, Spain. (Chapter 5)

- **Posters**

  - SDN DAYS 2017 and 2018 (Paris, France): Scalable and Cost-Efficient Algorithms for Resource Allocation Problems in C-RAN. **Poster and Talk**.

  - CLOUD DAYS'2017 (Nancy, France): Resource allocation and BBU split placement in Cloud Radio Access Networks **Poster and Talk**.

Chapter 2

C-RAN optimization: mathematical background and state-of-the-art

2.1 Introduction

Network operators are investigating new algorithms for resource allocation problems to enable and facilitate the deployment of C-RAN architecture. Combinatorial optimization is considered as one of the most efficient techniques to address such problems. In the first section of this chapter, we describe some combinatorial optimization concepts and algorithms that we use to propose exact and heuristic algorithms to optimally address resource allocation and network optimization problems in the context of C-RAN. We introduce also the homology theory which provides efficient algorithms to deal with coverage hole detection problems. In the second section, we provide a deep analysis of the most relevant approaches in the literature which have been proposed to address resource allocation problems in C-RAN.

2.2 Mathematical background

This section contains an overview of the most basic concepts that will be used in the following chapters. We address some fundamental concepts concerning network optimization with a focus on combinatorial optimization background and simplicial homology approaches.

2.2.1 Combinatorial optimization : techniques and algorithms

Combinatorial optimization techniques are very useful in modeling several types of problems such as planning, routing, scheduling, assignment, and design that appear in many real life applications. Most of these problems are very complex and thus

very hard to solve. In this section, we introduce some combinatorial optimization techniques and algorithms that we will use in this manuscript. We introduce linear programming and different methods used to solve optimization problems. Then, a class of approximation algorithms is presented to tackle scalability issues of some optimization problems. Finally, we describe some classical optimization problems and outline the most efficient algorithms proposed to optimally solve them.

2.2.1.1 Linear programming

Linear Programming (LP) is a powerful tool to model optimization problems by means of mathematical relations, allowing to find the best solution from all possible ones. In fact, LP is a mathematical formulation where problem decisions are represented by decision variables and problem constraints are expressed by mathematical relations that describe conditions imposing the feasibility of the solutions. The aim of LP is to find the best values of decision variables that satisfy all problem constraints and maximize (or minimize) an objective function. LP is a mathematical programming model in which the objective function is a **linear** expression of the decision variables and the constraints are given by a system of **linear** inequalities. Each LP can be represented by the following standard form:

$$\begin{aligned} & opt \quad c^T x \\ Subject\ to: \quad & Ax \leq b \\ & x \geq 0 \end{aligned} \tag{2.1}$$

- x : vector of **decision variables** which are the quantities to be determined in order to solve the problem.
- $opt \quad c^T x$: **objective function** is a function of decision variables that aims to maximize or minimize some numerical value. This value can represent profit, cost, revenue, distance, etc.
- $Ax \leq b$: **constraints** represent some mathematical relations to be respected by the final solution which are expressed by linear inequalities.
- c^T, A and b : are respectively matrix transpose of coefficients, matrix of coefficients and vector of coefficients.

Decision variables in the above LP (2.1) are continuous (real values). However, in some cases, these variables could be integer. This leads to two other variants of LP :

- Mixed Integer Linear Program (MILP) if only some of decision variables are integer while some others can take real/continuous values.

- Integer Linear Program (ILP) if all decision variables are integer. This variant is very well known and often used to model optimization problems that represent the integrity conditions of various real-life problems. The ILP formulation can be represented as follows :

$$\begin{aligned} & opt \quad c^T x \\ Subject\ to: \quad & Ax \leq b \\ & x \in \{0, 1\} \end{aligned} \tag{2.2}$$

Before providing an optimal solution, solving an ILP of (2.2) consists in exploring all possible solutions that satisfy all problem constrains. These solutions represent the set of feasible solutions and allow to constitute a **convex polytope** which is the region obtained by the intersection of incident vectors, each of which represents a constraint in the problem (as shown in Figure 2.1).

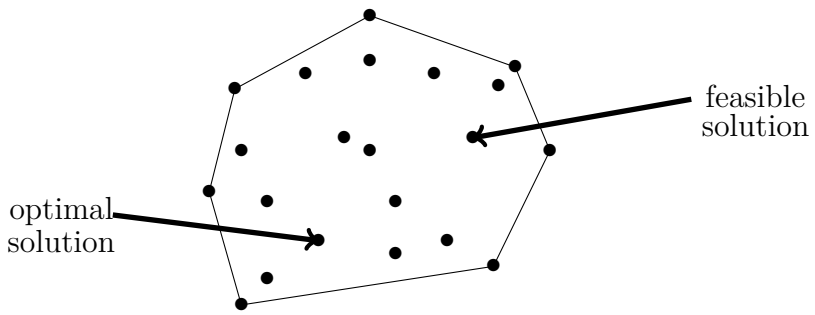


Figure 2.1: An example of a convex polytope

ILP models have many advantages, including :

- Solve optimization problems when guaranteeing optimum solution.
- Modeling very complex optimization problems by a simple mathematical formulation where all problem constraints are represented by linear inequalities and the goal of decision making is measured by cost (objective) function.
- A wide range of problems in real life can be modeled and solved easily using different LP models.

In the following, we focus on ILP models that we mainly use to formulate and solve the different addressed problems in this thesis.

2.2.1.2 Solving ILP : methods

Without loss of generalities, the simplest way to find an optimal solution for an optimization problem is to explore and evaluate all possible solutions and then select the **best one**. In this section, we introduce three methods that are used to solve an ILP : (a) Branch-and-Bound, (b) cutting plane, (c) Branch-and-Cut.

(a) Branch-and-Bound

Branch-and-Bound is the simplest approach to obtain an **optimum solution** of an ILP problem by exploring a complete enumeration of all possible solutions. Branch-and-Bound algorithm is based on the construction of **branching tree** in which the "root" node represents the original problem and each "son" node corresponds to a subproblem. This algorithm splits recursively each problem into two sub problems, explores different branches of the tree and calculates the solution value of the corresponding subproblem at each node. The main two steps of Branch-and-Bound algorithm are :

- **Branch**: is the process of creating new subproblems from an original problem. In fact, each problem (or subproblem) is partitioned into two subproblems such that the union of feasible solutions of the subproblems represents the feasible solutions of the original problem. This step is executed recursively.
- **Bound**: is the process of evaluating each node. In fact, for each node, we calculate the solution value obtained by the corresponding subproblem. We stop branching this node if the obtained value of the corresponding problem is above (in the case of minimizing problem) the best feasible solution found so far.

Branch-and-Bound algorithm is an exact method allowing to find the optimum solution. However, the number of nodes in the branching tree exponentially increases with problem size. Therefore, to reduce the number of nodes in the tree, some techniques such cutting plane are used to accelerate the convergence time to solve the problem.

(b) Cutting plane

Cutting plane method is a technique used to accelerate the search of optimal solutions for an ILP formulation. This approach considers the **LP relaxation**, where the integrity constraints $x \in \{0, 1\}$ are replaced by $x \in \mathbb{R}_+$. Cutting plane method consists in solving the LP relaxation and verifying if the obtained solution contains some fractional variables or not. If at least one variable is fractional, new valid inequalities, that are violated by these fractional variables, are investigated and then added to the current LP relaxation. These valid inequalities are called **cutting planes**.

The cutting plane method is always used with Branch-and-Bound algorithm allowing to reduce the space solution (convex polytope) of the optimization problem and thus find optimal solutions faster. The most well-known cutting plane algorithm is the Gomory Cut algorithm [37] which consists in finding the constraint to separate a fractional solution to any linear relaxation. Gomory algorithm is extended then by Gomory-Chvătal algorithm.

Nevertheless, finding new valid inequalities violated by the obtained solution (i.e. cutting planes) is not often very easy. In fact, the problem of finding such inequalities is called **separation problem** which can be defined as follows:

Given a solution $x^* \in \mathbb{R}^n$ lying outside of the polytope P, a separation problem consists in finding a linear inequality $ax \leq \alpha$ which is valid for the polytope and violated by the solution x^* (see Figure 2.2).

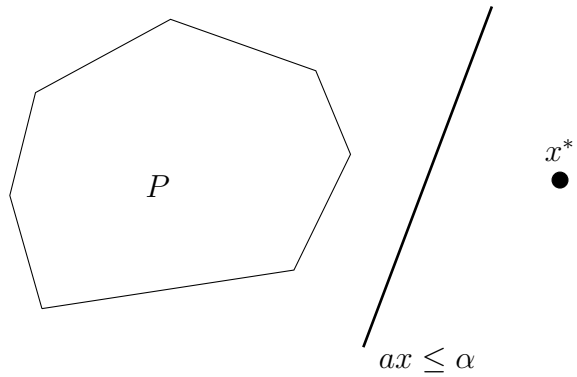


Figure 2.2: An example of linear inequality separating the convex polytope P and the solution x^*

(c) Branch-and-Cut

Branch-and-Cut is a method that combines Branch-and-Bound algorithm with cutting plane technique to solve ILP models. In fact, Branch-and-Cut considers the **LP relaxation**, where the integrity constraints of the ILP formulation (2.2) are relaxed (the constraints $x \in \{0, 1\}$ are replaced by $x \in \mathbb{R}_+$). The LP relaxation is then solved and a lower (resp. upper) bound, in the case of minimization (resp. maximization) problem, is found. If the obtained solution contains fractional variables, new cutting planes are investigated (as described above) and then added to the current LP relaxation. The whole process is iterated until all variables in the obtained solution are integers. We summarize in Algorithm 1 the Branch-and-Cut method.

2.2.1.3 Approximation algorithms

In addition to exact approaches based on ILP formulation, we will propose, later in this thesis, new approximation algorithms based on exact theories and approaches such as matroid and b-matching. The proposed heuristic algorithms can converge faster and scale to larger problem instances.

(a) Matroid

Matroid is a discrete structure that generalizes the concept of independence in linear algebra. There are several ways to define a matroid such as bases, the rank function, independent sets and cycles.

In the following, we will use the definition based on independent sets. We recall that in graph theory, an independent set is a set of edges in a graph that do not contain a cycle[38]. The definition of a matroid is provided by [39] and can be formulated as follows.

Algorithm 1 Branch-and-Cut algorithm

Input: An ILP formulation (2.2), denoted by ILP^0

Output: An optimum solution S^*

Set $L := \{ILP^0\}$, set $x := -\infty$, and set $x^* := +\infty$

for each ILP^l in L **do**
  Step 1. Solve the LP relaxation of ILP^l and calculate the value of obtained solution x

  Step 2.
  if $x < x^*$ **then**
    Update the value of the best known solution : $x^* := x$
  else
    Cut the node which contains the current problem and go to Step 1
  end if

  Step 3.
  if The obtained solution contains some fractional variables **then**
    Find the cutting planes that are violated by the obtained solution (some may be sufficient)
    Add the violated inequalities to the current LP relaxation
    Branch the current problem into two subproblems and put them into L
  end if
end for

Definition A matroid $M = (E, \mathcal{F})$ is a structure in which E is a finite set of elements and $\mathcal{F}$ is a family of subsets of E verifying the following principal properties:

(P1) $\emptyset \in \mathcal{F}$.

(P2) If $A \in \mathcal{F}$ and $B \subseteq A$, then $B \in \mathcal{F}$.

(P3) If $A, B \in \mathcal{F}$, and $\mid B \mid > \mid A \mid$ thus $\exists e \in B \setminus A$, such that $A \cup \{e\} \in \mathcal{F}$.

If $\mathcal{F}$ is only satisfying the properties (P1) and (P2), then we are invoking an **independent system**. A maximal set of E is said to be *basis* of matroid and the rank $r(A)$ of a subset of A is the cardinality of a maximal independent subset A. We note that all basis of a matroid have the same cardinality.

Other definitions and more details on matroid theory can be found in [25], [40] and [41], for instance.

There are many examples of matroids such as uniform matroid, linear matroid, graphic matroid. In the following, we introduce the graphic matroid that we will use to efficiently solve resource allocation problems.

Graphic matroid (also known as cycle matroids of a graph) : is a matroid whose independent sets are the forests in a given graph and can be represented as follows:
Given a graph $G = (V, E)$, a graphic matroid is $M = (E, \mathcal{F})$, in which $\mathcal{F}$ is a set of trees' forests in G : $\mathcal{F} = \{F \subseteq E \mid G_M = (V, F)$ *is the induced subgraph of* G *such that* F *is a forest*$\}$

In the case of the graphic matroid, the problem of finding the minimal (maximal) forest can be optimally solved by a simple greedy algorithm (see [40] and [42] for more details). This algorithm will be used with some modifications when involving resource allocation problem in Chapter 3.

(b) b-Matching

Let $G = (V, E)$ be an undirected graph with edge weights $u(e)$ for each edge $e \in E$ and node capacities $b(v)$ for each node $v \in V$. We denote by $\delta(v)$ the set of incident edges of v. A b-matching is a generalization of ordinary matching where all node capacities $b(v)$ are equal to one. We note that a matching in G is a set of pairwise disjoint edges (i.e. the endpoints are all different) and it is called a perfect matching if all vertices are covered. In the following, we introduce the definition of capacitated, i.e. the edge weights $u(e)$ are finite numbers, b-matching provided by [39].

Definition Let G be an undirected graph with integral edge capacities $u : E \to \mathbb{N} \cup \{\infty\}$ and numbers $b : V \to \mathbb{N}$. Then a b-matching in (G, u) is a function $f : E \to \mathbb{Z}_+$ with $f(e) \leq u(e)$ for all $e \in E$ and $\sum_{e \in \delta(v)} f(e) \leq b(v)$ for all $v \in V$. In the case $u = 1$ we speak of a simple b-matching in G. A b-matching is called perfect if $\sum_{e \in \delta(v)} f(e) = b(v)$ for all $v \in V$.

The b-matching polytope of (G, u) is the set of vectors $\boldsymbol{x} \in \mathbb{R}_{+}^{\boldsymbol{E}}$ satisfying :

$$\begin{cases} x_e \leq u(e), \quad \forall e \in E & (2.3a) \\ \sum_{e \in \delta(v)} x_e \leq b(v), \quad \forall v \in V & (2.3b) \\ \sum_{e \in E(G[X])} x_e + \sum_{e \in F} x_e \leq \lfloor \frac{1}{2}(\sum_{v \in X} b(v) + \sum_{e \in F} u(e)) \rfloor, & \\ \qquad \forall X \subseteq V(G), F \subseteq \delta(X) & (2.3c) \end{cases}$$

where $E(G(X))$ represents a subset of edges in the subgraph $G(X)$ generated by a subset of vertices X and $\delta(X)$ is a set of incident edges of X.

Constraints (2.3c) represent the blossom inequalities which are very used to model several **NP-Hard** optimization problems and find their convex hulls. In [43], authors gave a polynomial-time separation algorithm for b-matching polytopes by identifying the violated blossom inequalities. In [44], authors proved that the separation problem for b-matching polytope (in the case of capacitated b-matching) can be solved in **polynomial time :** $\boldsymbol{O}\left(|\boldsymbol{n}|^{\boldsymbol{2}}|\boldsymbol{m}| \ln\left(\frac{|\boldsymbol{n}|^{\boldsymbol{2}}}{|\boldsymbol{m}|}\right)\right)$ where n and m are the number of vertices and edges respectively in the graph G.

2.2.1.4 Combinatorial optimization problems : some examples

We present in the following some combinatorial optimization problems that will be used in this manuscript.

(a) Shortest path problem

One of the well-known combinatorial optimization problems is the problem of finding a shortest path between two specified vertices in a graph G such that the total sum of the edges' weights is minimum. G can be directed or undirected graph and can contain negative weights. The shortest path problem can be solved in **polynomial time** if there are no negative cycles in G [45].

There are three variants of shortest path problem :

- Two nodes shortest paths: given two nodes s and t, find the shortest path between s and t
- Single source shortest paths : given a node s, compute the shortest paths from a node s to all vertices in the graph.
- All pairs shortest paths: find a shortest path for all ordered pairs of vertices (s, t) in the graph.

Many algorithms have been proposed to solve these variants of shortest path problem. One of the most used algorithms is Dijkstra algorithm [46] which consists in finding shortest paths in a graph (directed or undirected)

from a source s to all other nodes, instead of just a specific pair of source and destination nodes. This is very useful in telecommunication networks where we always need to compute the shortest path from the source to all destinations. We note that the complexity of this algorithm is $\boldsymbol{O(n^2)}$ where n is the number of vertices in G.

Unlike Dijkstra algorithm that can be applied on a graph with only non-negative edge weight values, Bellman-Ford algorithm [47] was proposed to find shortest paths in a graph which may contain negative weights on edges. For further reading about shortest path algorithms, see [48] for instance.

(b) Maximum flow problem

Let $G = (V, E)$ be a directed graph with edge capacities $c : E \to \mathbb{R}_+$ and two specified vertices, s source and t sink. The triple (G, s, t) is called a network. The problem of maximum flow consists in transporting simultaneously as many units as possible from s to t and can be defined as follows.

Definition A network flow of (G, s, t) is a function $f : E \to \mathbb{R}_+$ which satisfies the following proprieties:

- $f(x, y) = -f(y, x), \quad \forall (x, y) \in E$
- $f(x, y) \leq c(x, y), \quad \forall (x, y) \in E$ where $c(x, y)$ is the capacity on the edge (x, y)
- $\sum_{y \in V} f(x, y) = 0, \quad \forall x \in V \backslash \{s, t\}$

The problem of finding a maximum flow in G can be solved in **polynomial time** [49] and many algorithms have been proposed such as Ford-Fulkerson algorithm [46] which has a complexity of $\boldsymbol{O(|m|^2|n|)}$ where n and m are the number of vertices and arcs respectively in G.

(c) Knapsack problem and multiple knapsack formulation

Another well-known combinatorial optimization problem is the knapsack problem which consists in finding an optimum subset from a set of items to be filled into a knapsack with limited capacity. In fact, given a knapsack with a maximal capacity c and a set of items j, each of which has a profit p_j and a weight w_j. Knapsack problem aims to find a subset of items such that the total profit of the selected items is maximized and the total weight does not exceed the capacity of knapsack C. Alternatively, knapsack problem can be formulated by the following ILP :

$$\begin{aligned} &\max \quad \textstyle\sum_{j=1}^{n} p_j x_j \\ &S.T. : \\ &\textstyle\sum_{j=1}^{n} w_j x_j \leq C; \\ &x_j \in \{0, 1\}, \quad \forall j = 1, ..., n; \end{aligned} \tag{2.4}$$

The knapsack problem is **NP-Complete** [50] but there is a pseudopolynomial algorithm which can be quite efficient if the involved numbers (p_j and w_j) are not too large [51].

In the following, we provide a knapsack algorithm based on dynamic programming approach [52] which can find optimal solutions in $O(nP)$ time, where n is the number of items and $P = \sum_{j=1}^{n} p_j$ [51]. The algorithm 2 will be used in Chapter 3 when involving a resource allocation problem.

Algorithm 2 Dynamic programming algorithm for knapsack problem

Input: Non-negative integers : n items, a profit p_j and a weight w_j for each item j, a knapsack with capacity C

Output: An optimal subset of items to be filled into the knapsack

Step 1. Let P be an upper bound on the value of the optimum solution : $P = \sum_{j=1}^{n} p_j$

Step 2. Set $x(0,0) := 0$ and $x(0,k) := \infty$ for $k = 1, ..., P$

Step 3.
for $j := 1$ to n **do**
  for $k := 0$ to P **do**
    Set $s(j,k) := 0$ and $x(j,k) := x(j-1,k)$
  end for
  for $k := p_j$ to P **do**
    if $x(j-1, k-p_j) + w_j \leq \min\{C, x(j,k)\}$ **then**
      Set $x(j,k) := x(j-1, k-p_j) + w_j$ and $s(j,k) := 1$
    end if
  end for
end for

Step 4. Let $k = \max\{i \in \{0, ..., P\} : x(n,i) \leq \infty\}$, Set $S := \varnothing$
for $j := n$ down to 1 **do**
  if $s(j,k) = 1$ **then**
    Set $S := S \cup \{j\}$ and $k := k - p_j$
  end if
end for

We introduce in the following the definition of multiple knapsack formulation according to [30] that we use to propose new resource allocation algorithm in Chapter 4.

Definition Given a set of n items and a set of m knapsacks ($m < n$), with p_j = profit of item j, w_j = weight of item j, c_i = capacity of knapsack i, find m disjoint subsets of items with the total profit of the selected items is a maximum, and each subset can be assigned to different knapsacks whose capacity is less than the total weight of items on the subset.

The multiple knapsack problem is a generalization of the knapsack problem from a single knapsack to m knapsacks, each of which has limited capacity. The objective of multiple knapsack problem is to assign each item to at most one of the knapsacks such that none of the capacity constraints are violated and the total profit of the items putted into knapsacks is maximized.

In addition to combinatorial optimization techniques that we mainly use to propose new algorithms in this thesis, we briefly introduce in the following the homology theory that provides powerful solutions to address network coverage hole detection problems in Wireless Sensor Networks (WSN). In fact, we focus on two simplicial homology-based approaches [33] : **Čech complex** and **Rips complex**, which are very close to our proposal to model the full network coverage problem in the context of C-RAN. Furthermore, we will use these approaches to benchmark our proposed algorithm (see Chapter 4).

2.2.2 Simplicial homology background

One of the most efficient approaches to deal with coverage hole detection problem is simplicial homology [33]. In fact, homology based approaches consist in analyzing the topological properties of a domain or region by algebraic computations. Instead of graphs, more generic objects are used, known as simplicial complexes. In this section, we introduce the definition of a simplicial complex and we describe two most useful approaches based on simplicial homology : **Čech complex** and **Rips complex** (see [33] for more details about both approaches). These approaches will be used in Chapter 4 to evaluate the performance of our proposed algorithm when dealing with the full network coverage problem in C-RAN. For further details about simplicial homology theory, see [53] for instance.

2.2.2.1 Simplicial complex and k-simplex

A simplicial complex is a combinatorial object composed by vertices, edges, triangles, tetrahedra, and their $n-$dimensional counterparts, each of which represents $k-$simplex. $k-$simplex is an unordered subset of $k+1$ vertices, where k is the dimension of the simplex.

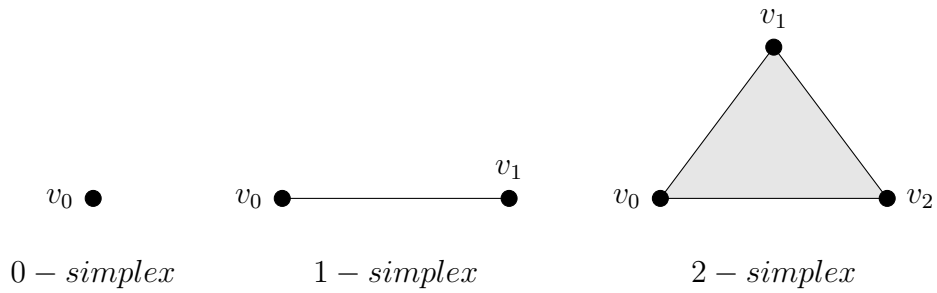


Figure 2.3: Example of k-simplices

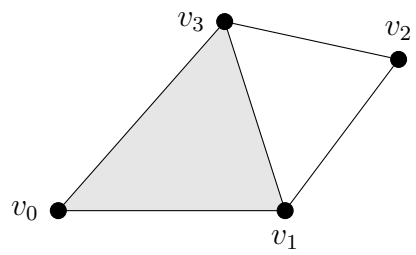


Figure 2.4: Example of simplicial complex

In fact, we show in Figure 2.3 some examples of k-simplex : 0-simplex is a vertex, 1-simplex is an edge, 2-simplex is a triangle. To better understand these notions, we present, in Figure 2.4, a simple example of a simplicial complex that contains four 0-simplices : $\{v_0\}$, $\{v_1\}$, $\{v_2\}$ and $\{v_3\}$, five 1-simplices $\{v_0, v_1\}$, $\{v_0, v_3\}$, $\{v_1, v_2\}$, $\{v_1, v_3\}$ and $\{v_2, v_3\}$ and one 2-simplex : $\{v_0, v_1, v_3\}$ while $\{v_1, v_2, v_3\}$ is not 2-simplex.

Homology techniques are very well used to model the topology of wireless sensor networks and detect the existence of coverage holes. In the following, we introduce two approaches, Čech complex and Rips complex, which use the simplicial homology to verify the connectivity of the network and detect the existence of coverage holes. In fact, for these approaches, the construction of simplicial complex enables to detect and count the number of $k-$dimensional holes, called Betti numbers. Indeed, the $k-th$ Betti number, denoted by β_k (for $k-$dimensional hole), counts the number of cycles of $k-$simplices which are not filled with $(k+1)-$simplices. For connectivity and coverage problems, only the first two Betti numbers are considered :

- β_0 : indicates the number of $1-$dimensional holes, that is the number of connected components in the simplicial complex (for example, $\beta_0 = 1$ in simplicial complex in Figure 2.4).

- β_1 : indicates the number of $2-$dimensional holes, that is the number of holes in the simplicial complex (for example, $\beta_1 = 1$ in the simplicial complex of Figure 2.4) .

2.2.2.2 Čech complex and Rips complex

We focus in this section on the two most useful approaches based on simplicial homolgy, which are Čech complex and Rips complex. These approaches are based on the verification of intersection between cells, to detect holes and connectivity problems. Čech complex is introduced in [33] and can be defined as follows :

Theorem 2.2.1 *Čech complex of a set of points U, Č(U), is the abstract simplicial complex whose $k-$simplices correspond to nonempty intersections of $k+1$ distinct elements of U, i.e. $[v_0, v_1, ..., v_k]$ is $k-$simplex if and only if $(v_0) \cap (v_1) \cap ... \cap (v_k) \neq \varnothing$, Where (v_i) is the cell centered at v_i.*

Unfortunately, it is very difficult to compute Čech complex due to its high complexity. Hence, another approach based on simplicial complex, named Rips complex, is introduced to deal more easily with coverage and connectivity problems. It is defined as follows.

Theorem 2.2.2 *Rips complex of a set of points U, $R(U)$, is the abstract simplicial complex whose $k-$simplices correspond to unordered $(k+1)-$tuples of points in U which pairwise intersect, i.e. $[v_0, v_1, ..., v_k]$ is $k-$simplex if and only if $(v_i) \cap (v_j) \neq \varnothing$, Where (v_i) is the cell centered at v_i.*

Comparison between Čech complex and Rips complex

According to theorem 2.2.1, Čech complex provides the exact topology of the network allowing to verify the connectivity of the network (β_0) and compute the number of existing holes (β_1). However, it is very difficult to compute the Čech complex due to the high complexity of determining k-simplices. Rips complex can be constructed based on the connectivity graph of the network and gives an approximate coverage by simple algebraic calculations. In [54], authors prove that Rips complex provides a solution with an error between 0.5% and 3%. Hence, in Chapter 4, we will use the Rips complex-based approach to model the full network coverage problem in the context of C-RAN and to benchmark our proposed algorithm using different simulation scenarios and performance metrics.

2.3 C-RAN optimization : state-of-the-art

Combinatorial optimization techniques and simplicial homology theory are the main techniques that we will use to address three exciting research challenges in the context of C-RAN. These challenges will be detailed in this chapter with a deep analysis of the state-of-the-art on the most relevant researches in the literature.

2.3.1 Constrained resource allocation problem

The deployment of C-RAN architecture, where the infrastructure is shared across multiple cell sites, is expected to reduce both capital and operating expenditures (CAPEX and OPEX) as well as to improve the resource utilization efficiency [55]. Such gains can only be achieved by efficiently assigning the antennas (RRHs) demands to the centralized data centers (BBU pool) when latency and processing requirements are met. Therefore, to address this constrained resource allocation problem, network operators are investigating new algorithms to determine the best strategies to assign RRHs to BBUs (known as RRH-BBU assignment problem). The proposed algorithms will jointly assign the processing and radio resources to antennas demands taking advantage of the computing resource pooling in common edge data centers. The optimal mapping between RRHs and BBUs, i.e. optimal RRH-BBU assignment, is reached when jointly minimizing the communication latency on the fronthaul network and computing resource consumption.

In this context, authors in [56] and [57] discussed new mathematical modeling to cope with RRH-BBU assignment problem. They proposed a mathematical model based on an ILP approach in which only BBUs processing capacity constraints are considered. The proposed exact optimization model does not take into account the transmission delay on the fronthaul network and the latency requirements of antennas demands. To cope with scalability issues, both these references proposed approximation algorithms that do not guarantee the convergence to an optimal solution. In this thesis, we address the RRH-BBU assignment problem when jointly meeting the strong latency requirements on fronthaul network and the edge data centers' limited capacity constraints. Our joint optimization is represented by an exact formulation before investigating heuristic algorithms that converge to near-optimal solutions in acceptable times.

Authors of reference [58] proposed a load-aware dynamic mapping between RRHs and BBUs with the aim of minimizing the number of active BBUs required to process the computational resource demands. The authors introduced a heuristic DRA for Dynamic RRH Assignment to dynamically optimize the BBU pooling gain. They claim that their approach delivers an almost optimal performance in terms of computational resource gain and convergence time as compared to First-Fit Decreasing (FFD)[1] algorithm. Similarly, another resource allocation algorithm was introduced in [59] to minimize the number of active BBUs required to serve all users in the network to save more energy. In this manuscript, and in addition to the proposed ILP algorithm used as reference to benchmark other approaches, we propose three heuristic approaches to guarantee the convergence of the constrained resource allocation problem to optimal solutions in negligible times.

Another work addressing the RRH-BBU assignment and resource allocation problem is proposed in [60]. Indeed, the authors of this reference proposed a greedy algorithm to assign the aggregated demands of each cell to the BBU pool in such a way that the power consumption of the physical resources is minimized. The authors did not consider the latency requirements in their optimization model. Since the latency and the transmission delay constraints are very strong in C-RAN architecture, we propose exact and heuristic algorithms based on a joint optimization of communication latency and computing resource allocation.

In [61], the authors introduced a mathematical formulation based on ILP to optimally assign antennas demands to different BBU pools. This work aims to minimize the length of fiber while maximizing the statistical multiplexing gain for each BBU pool hosting the baseband functions. Their approach shows that the optimal assignment of RRHs to the BBU pools depends on the length of fiber and BBU resources. In our work, we propose an exact formulation for the same problem and to scale, our contribution consists in investigating new and rapid approaches to guarantee the convergence to near optimal solutions when considering the same parameters than those used in [61].

[1]FFD sorts all items in decreasing order of their sizes, and then puts each item into the first bin that has sufficient remaining space.

Authors in [62] investigated new algorithms to determine the best strategies for RRH-BBU mapping by finding the optimal clustering of existing RRHs. They modeled as bin packing problem when considering two main constraints (i) the radio resources of each active BBU must be enough to meet the demands of its mapped RRHs and (ii) the set of antennas, that will be assigned to each BBU, should be geographically adjacent. Exact and heuristic algorithms are provided to reduce network power consumption when guaranteeing good QoS for end-users. Nevertheless, the proposed formulation did not consider the communication latency on the fronthaul network joining RRHs to BBU pools. In this manuscript, we address the RRH-BBU mapping problem by proposing an exact approach based on ILP model and approximation algorithms to find the best assignment of antennas to centralized data centers when jointly considering the limited processing capacity in BBU pools and the transmission delay on fronthaul links.

Similarly to [62], authors in [63] formulated the problem of RRH-BBU assignment as a bin packing problem. In fact, after proposing an ILP model to address this problem, authors in [63] used a simple Best Fit Decreasing (BFD)[2] algorithm to assign RRHs to BBUs and then determine the number of active BBUs that should be used to meet antenna demands (BFD is a well-know algorithm developed by [64] to solve bin packing problem). In our work, in addition to an exact approach based on ILP formulation, we propose new approximation algorithms to find near-optimal solutions to deal with RRH-BBU assignment problem in acceptable times. These algorithms will be benchmarked with the exact approach using different simulation parameters and according to many performance metrics.

Some existing works (for instance [65], [66] and [67]) addressed the resource allocation problem in C-RAN by only focusing on minimizing the energy consumption in the BBU pool without taking into account the fronthaul latency constraints. In this manuscript, we seek new algorithms to reduce the network costs, i.e. CAPEX and OPEX, by jointly optimizing the resource consumption and the communication latency in order to achieve optimal utilization of processing resources.

2.3.2 Network coverage optimization problem

In the context of C-RAN architecture, network operators are seeking to increase the density of existing cells by deploying more antennas in order to enlarge the network spectrum and enhance the network coverage and capacity [68]. Nevertheless, this brings new challenges in inter-cell interference management and reduction when maintaining a full network coverage. Indeed, to jointly cope with these goals, numerous schemes have been proposed in different networks, such as Wireless Sensor Networks (WSN), using different approaches.

One of the most used methods to detect coverage holes in WSNs is Delaunay triangulation which consists in dividing the target field into triangles that have

[2]BFD sorts the items to be inserted in decreasing order of size, and puts each item into the fullest bin in which is fits.

no other nodes inside. In Figure 2.5, we represent an example of graph based on Delaunay triangulation method.

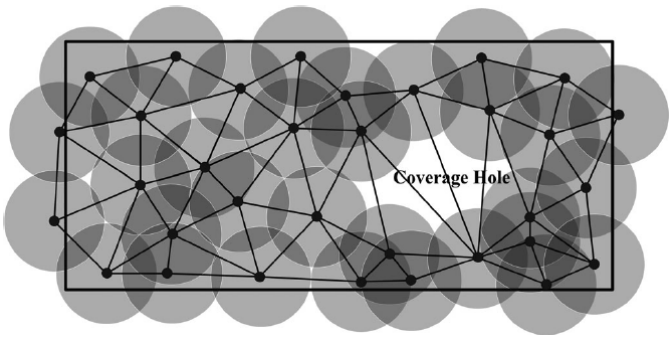


Figure 2.5: Delaunay triangulation. Source: [2]

Authors in [2] proposed a mathematical model based on Delaunay triangulation to detect coverage holes in WSN and found the shortest paths for node movement to heal the holes. The proposed algorithm used the Delaunay triangulation to find a necessary and sufficient condition to determine the coverage of a triangle that has no other nodes inside its circumcircle. The time complexity of this method is close to $O(bn)$ where n is the total number of nodes and b represents the number of adjacent nodes in the vicinity of each node. However, this proposal is based on a simple mathematical formula that does not address all of the possible scenarios. Furthermore, the proposed algorithm requires that cells should be identical with the same coverage radius. In our considered C-RAN architecture, the antennas (cells) have different coverage radius, depending on number of users supported by this antenna. Therefore, we need to investigate new approaches to optimize the network coverage when considering different cells' radii.

Other works addressed the coverage problem using probability methods. In [69] and [70], the authors used a probability method to eliminate all coverage holes detected in the considered network by determining the smallest size of cells. Similarly to [2], authors of these papers supposed that all cells have the same coverage radius which is not realistic in the context of C-RAN. In our work, we propose new cutting planes in Branch-and-Cut algorithm to rapidly detect coverage holes and reduce inter-cell interference jointly.

Authors in [71] proposed an ILP model that maximizes the total coverage in the context of WSN. The main limitation is that they discretize the area to be covered into several cells, and each cell is discretized into several points. In the context of C-RAN, we consider distributed antennas with different coverage radius and we aim to optimize the total network coverage by adjusting the antennas radii.

Other works used simplicial homology (detailed in Section 2.2.2) to address the coverage hole detection problem. In [72], the authors proposed an heuristic algorithm to turn off the minimal number of cells without generating coverage holes. In

fact, authors used the simplicial complex which is introduced in [33] and [73] as a representation of coverage topology. In this algorithm, at every time a cell is turned-off, we need to compute the Betti numbers (β_0 and β_1 as defined in Section 2.2.2) to ensure that the network coverage is maintained. However, turning off cells can not optimize neither network coverage nor overlapping region. In this manuscript, we propose an exact mathematical formulation to provide optimal solutions for network coverage problem.

Similarly, another algorithm based on simplicial homology has been introduced in [74]. This algorithm aims to minimize the total consumed power for wireless networks. They used a heuristic approach based on Simulated Annealing (SA) to find sub-optimal solutions instead of investigating rapid and efficient approaches to attend optimal solutions. The SA algorithm adjusts the coverage radius of each cell by building a complex graph based on RIPS complex and then computes the Betti numbers to avoid the generation of coverage holes. In this manuscript, we propose an optimization model considering multiple objectives such as inter-cell interference reduction and network coverage holes elimination.

Other works tackled the network coverage problem in the context of C-RAN when addressing multi-dimensional resource optimization issues. In fact, authors of [75] proposed a Cloud-Based Radio over Optical Fiber Network (noted by C-RoFN) architecture with multi-stratum resources optimization using Software Defined Networks (SDN) paradigm to better get a grasp of resource optimization problems for C-RAN architecture. In the proposed architecture, optical spectrum and BBU processing resources are optimized jointly to maximize radio coverage when meeting quality of service. Authors in [76] provided a deep study on multi-dimensional resources integration for service provisioning in cloud radio over fiber networks. Indeed, they proposed a global optimization when considering together radio frequency, optical network and processing resources leading to maximize radio coverage. A mathematical modeling is then provided and an experimental test bed is used to confirm the efficiency and the feasibility of the proposed C-RoFN architecture.

2.3.3 BBU functions split and placement problem

The key obstacle in the adoption of C-RAN architecture, where the computing resources (BBUs) are decoupled from cell sites (RRHs), is the high-bandwidth constraints and low-latency requirements on the fronthaul network which connects the antennas (RRHs) to the edge data centers (BBU pools). In fact, authors in [7] investigate the cloudification of RAN by characterizing baseband processing times under different conditions. They underline that C-RAN architecture should take into account the fronthaul capacity constraints, the latency requirements for baseband processing and finally the execution environment (servers and operating systems). The strong latency requirements on the fronthaul network could be reduced by splitting the processing of baseband functions between RRHs and BBUs. In this context, several functional split options have been proposed by different organizations, e.g. NGMN [77] and 3GPP [78], in order to find good trade-off between BBU functions

centralization and fronthaul network requirements [77].

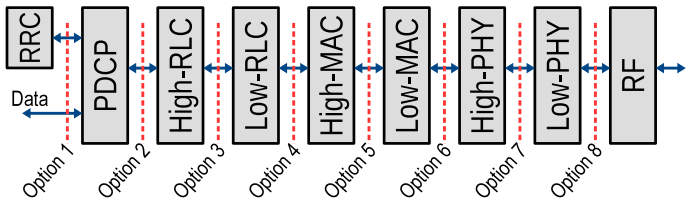


Figure 2.6: 3GPP functional split options

The 3GPP split proposal is shown in Figure 2.6, in which each functional split option represents a separation point between the layers that will be located in the cell sites (RRHs) and those located in the BBU Pools (centralized data centers). We recall that the main motivation to split the processing of BBU functions between RRH and BBU is to achieve the largest possible extent of centralization that a network architecture can allow.

Among the different functional split options depicted in Figure 2.6, our study focuses on the following four options which are illustrated in Figure 2.7 for a better understanding.

- Split 1 represents the traditional RAN architecture where PHY, RLC, MAC and PDCP functions are co-located in the cell sites (option 1 in Figure 2.6).
- Split 2 is a partial centralization of BBU functions where all functions in PHY, MAC and RLC layers are located in the cell sites and PDCP layer in the centralized data center (Option 2 in Figure 2.6).
- Split 3 is a partial centralization where PHY layer are located in RRHs and all functions in MAC, RLC and PDCP layers are incorporated in BBU pool (Option 6 in Figure 2.6).
- Split 4 represents fully-centralized architecture in which PHY, MAC , RLC and PDCP functions are moved to the BBU pools for centralized processing (Option 8 in Figure 2.6).

In the split 1, all functions in PHY, RLC, MAC and PDCP layers are located in the cell sites (RRH). Consequently, the latency constraints are less stringent which decreases its dependency on expensive fronthaul technology. However, benefits of virtualization and centralized multi-cell processing will be less. Contrary to split 1, split 4 represents the fully-centralized architecture, considered in the first C-RAN configuration, in which all functions in PHY, MAC, RLC and PDCP layers are moved to the BBU pool for centralized processing. Authors in [13] showed that the main benefit of this BBU function split option is its flexibility to support resource sharing and its efficiency to reduce energy consumption, while the disadvantage is the high data rate in fronthaul links expected for processing heterogeneous demands from multiple RRHs. In [79], authors highlighted, using a real test bed, the efficiency of the split option 4 in finding good trade-off between the processing of BBU functions in centralized data centers and the fronthaul requirements. In [80], authors

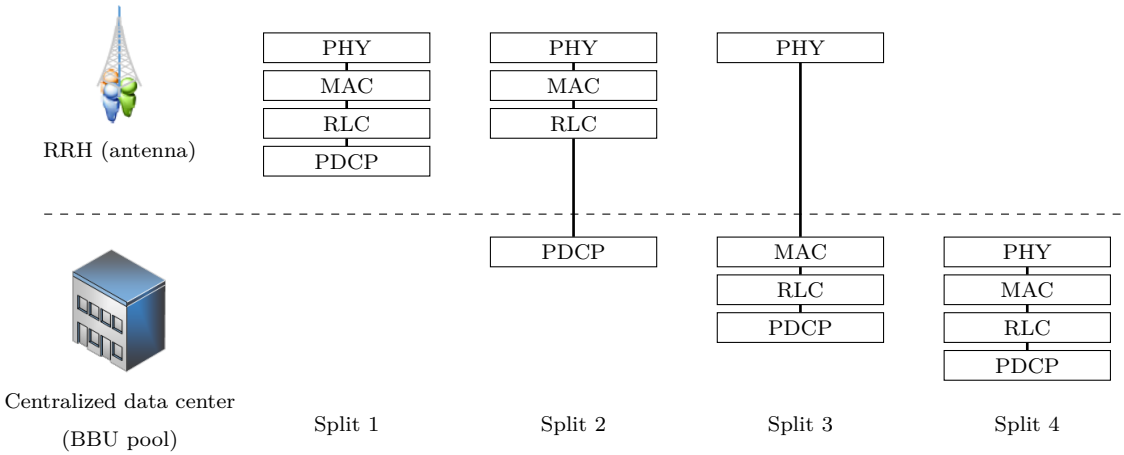


Figure 2.7: Overview of considered functional split options

confirmed that the maximum multiplexing gain on BBU resources can be achieved when considering the fully centralized C-RAN (split 4) after analyzing the different function splits. Nevertheless, split 4 requires high fronthaul network capacity compared to other split options such as Split 2. In fact, in Split 2, the PHY, MAC and RLC layers are integrated into RRHs while the PDCP layer is moved to the BBU pool to achieve increased multiplexing gains. This is feasible since functions in PDCP layer are less capacity-intensive and not subject to real-time constraints. Another BBU function split option that does not require high capacity on fronthaul network is Split 3 that consists in locating the PHY layer in the cell sites and moving other functions of upper layers to the centralized data centers. Obviously, both split options 2 and 3 have less fronthaul throughput compared to the split 4 in which the processing of baseband functions is centralized in BBU pool. However, the inter-cell collaborative processing cannot be efficiently supported when considering split 3 due to the the difficulty of interconnection between PHY layers and other layers.

Based on these different functional split options, many works investigated new approaches to determine the optimal placement of baseband processing functions in C-RAN architecture. In [81], authors proposed a low-cost solution by considering dual-site processing where baseband functions are divided between the antenna (or radio unit) and the data center (or digital unit). They provided an estimation of processing and bandwidth requirements for each functional split option and the total costs of ownership including equipment, civil work, commissioning as well as operational costs such as electricity and maintenance. The authors of [81] claimed that there is no "one-size-fits-all" solution for functional split and they indicated that hybrid RAN deployments contribute to Total Cost of Ownership (TCO) savings when considering 5G configurations. Authors in [82] proposed a methodology to derive guidelines for minimizing capital expenditure of C-RAN due to deployment of fronthaul and baseband processing resources. This work minimizes the total length of fiber while maximizing the multiplexing gain for each shared edge cloud hosting the baseband functions. The authors in [83] proposed a model representing the baseband processing functions as a directed graph where nodes represent different

processing functions and arcs represent connectivity between them. Then, they introduced the placement of these functions on data centers located at different sites. Computational costs and transport (link) costs are defined as well as constraints on the delay (latency) incurred both for processing and transport. In this paper, the problem of finding optimum locations to place baseband functions is equivalent to finding an optimum clustering scheme for the graph nodes with the objective of minimizing the total cost while ensuring that latency constraints are met.

2.4 Conclusion

In this chapter, we introduced some combinatorial optimization techniques and algorithms which will be used to address optimization problems in the context of C-RAN. First, we provided an overview of linear programming models which are considered as powerful techniques to obtain optimal solutions. Then, we described some optimization techniques that we used to deal with the addressed problems in this thesis and propose new heuristic algorithms to have good solutions in reasonable times even for large problem instances. Furthermore, we highlighted some well-known optimization problems and different proposed approximation algorithms. Another domain is represented in the second part which is the homology theory used to propose new solutions for network coverage problem in the context of C-RAN.

In the second section, we presented a deep analysis of the challenges that we will address in the following chapters and we highlighted the different approaches proposed in the literature.

In the next chapter, we will address an exciting research challenge in the context of C-RAN which is the assignment of antennas to edge data centers under transport requirements and limited capacity constraints and we will investigate new approaches to efficiently solve this problem.

Chapter 3

Constrained resource allocation in C-RAN

3.1 Introduction

The deployment of C-RAN architecture, where computing resources in edge data centers (BBU pool) are shared between multiple cell sites (RRHs), is expected to reduce network costs, e.g. CAPEX and OPEX, and improve the resource utilization efficiency. To achieve these goals, network operators need to investigate new efficient resource allocation algorithms to assign the limited processing resources in edge data centers to antennas (RRHs) demands when meeting strong latency requirements. The optimal mapping between RRHs and BBUs (RRH-BBU assignment) is achieved while jointly minimizing the communication latency on the fronthaul network and the resource consumption by reducing the number of active edge data centers needed to meet antennas demands.

For this purpose, an exact mathematical model based on Integer Linear Programming (ILP) is formulated to address the constrained resource allocation problem and provide the most appropriate strategies for RRH-BBU assignment when processing and latency requirements are met. Then, we seek new approximation algorithms that scale well, converge reasonably fast and find good solutions for RRH-BBU assignment problem.

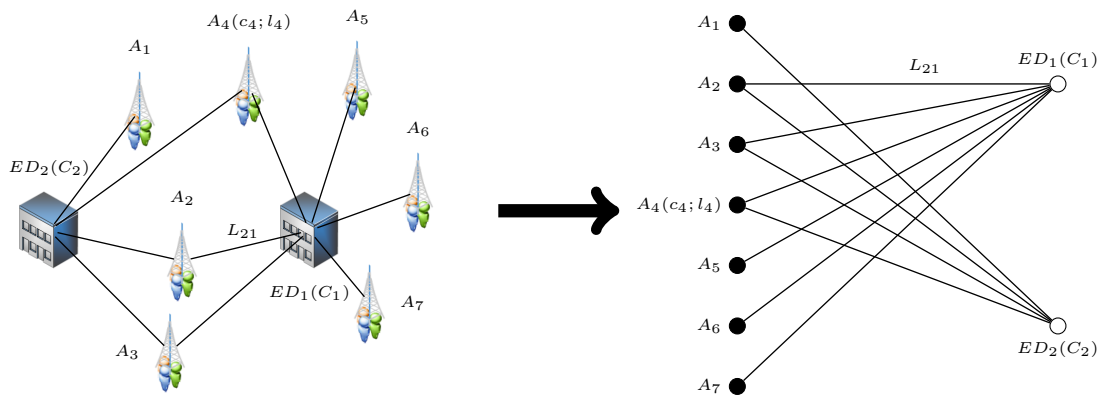
The remainder of this chapter is organized as follows. In Section 3.2, we describe the system model that will be used to define the constrained resource allocation problem and then we study the complexity of the addressed problem. In Section 3.3, an exact mathematical formulation is provided to meet the RRH-BBU assignment problem for small and medium size networks and then we introduce three approximation algorithms with significantly less complexity to deal with large problem instances. Numerical results are presented in Section 3.4 to highlight the performance of our proposed algorithms using several simulation scenarios and a real cellular network. Section 3.5 concludes this chapter.

3.2 Problem statement

In this section, we describe the system model that we consider to address the constrained resource allocation problem (RRH-BBU assignment problem[1]) and we introduce all variables and parameters used in the description of the problem. Then, and before providing our proposed algorithms, we discuss the complexity of the RRH-BBU assignment problem when considering all constraints that will be defined below.

3.2.1 System model

We consider the system model, as shown in Figure 3.1, to define the constrained resource allocation problem that aims to efficiently assign the antennas demands to the most appropriate edge data centers when strict latency and processing requirements are met. Our system model represents a C-RAN network where RRHs (antennas) and BBU pools (edge data centers) are deployed in a large area. As depicted in Figure 3.1a, our network architecture contains a set of antennas, denoted by I, each of which is defined by a position on the plane. These antennas $i \in I$ have variable expected latencies l_i and processing requirements in terms of CPU cores c_i, depending on aggregated end-users' demands. The RRHs are served by a finite set of available edge data centers denoted by J. Each edge data center $j \in J$ has a limited computing processing capacity C_j expressed as number of CPU cores.



(a) Network topology example (b) bipartite graph construction

C_1 (resp. C_2) : total number of available CPU cores in BBU pool ED_1 (resp. ED_2)
c_4 : number of CPU cores requested for processing the demands of antenna A_4
l_4 : expected latency for processing the demands of antenna A_4
L_{21} : communication latency on the fronthaul link between A_2 and ED_1

Figure 3.1: System model for constrained resource allocation problem

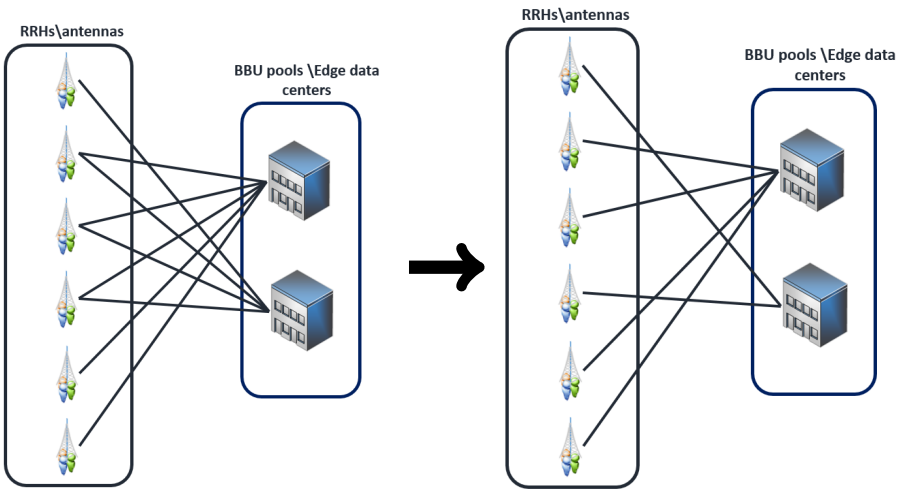
The antennas are connected to the edge data centers via fronthaul network which is represented by a set of communication links. Each fronthaul link between an

[1]We use "constrained resource allocation problem" and "RRH-BBU assignment problem" interchangeably throughout this chapter

antenna $i \in I$ and an edge data center $j \in J$ has a transmission delay L_{ij} that should be kept below **1 millisecond** in order to meet HARQ[2] requirements (see [11], [17] and [84]). This requires that the maximum distance d_{ij} between RRH i and BBU pool j must not exceed **20** to **40 kilometers** (see for instance [11] and [85]). The data traffic on the fronthaul network can be transmitted using different protocols, most commonly CPRI [9], or in some cases OBSAI [8]. In our system model, and according to [11] and [86], the transmission delay on the fronthaul network is **5 microseconds per Kilometer** and thus the communication (fronthaul) latency between RRHs and BBU pools vary between **100** and **200 microseconds** at the most.

As depicted in Figure 3.1, our network topology (Figure 3.1a) can be modeled by a weighted bipartite graph $G = (I \cup J, E)$ containing a set of antennas I in one side, a set of edge data centers J in the other side and a set of fronthaul links represented by the set of edges E. The weight value, denoted by L_{ij}, on each edge in the graph G represents the communication latency between the antenna $i \in I$ and the edge data center $j \in J$. The bipartite graph $G = (I \cup J, E)$ will be used to efficiently assign each antenna **to exactly one edge data center** when meeting the processing and latency requirements.

For sake of clarity, we give in Figure 3.2 a simple example of C-RAN network which is composed by 6 RRHs (antennas), 2 edge data centers (BBU pools) and a fronthaul network represented by a set of communication links. The constrained resource allocation problem consists in determining the optimal strategies to assign the antennas demands to the available edge data centers under strict processing and latency requirements.



(a) Initial graph (b) Final graph : a feasible solution

Figure 3.2: A solution example of the constrained resource allocation problem

[2]HARQ (Hybrid Automatic Repeat reQuest) is the process that poses the most stringent delay requirement for cellular networks

Hence, we aim to select, in the bipartite graph of Figure 3.2a, the optimal matching of all considered antennas with the available edge data centers when jointly meeting processing and latency requirements. The optimal assignment of all considered antennas to the BBU pools is achieved when latency and resource consumption (number of used edge data centers) are minimized. The right graph (Figure 3.2b) represents a feasible solution of the RRH-BBU assignment problem.

For sake of clarity, we summarize in Table 3.1 all variables and parameters that will be used, in the following, to model the constrained resource allocation problem.

Table 3.1: RRH-BBU assignment problem : variables and parameters

$G = (I \cup J, E)$	:	weighted bipartite graph
I	:	set of antennas/RRHs
J	:	set of edge data centers/BBU pools
E	:	set of communication links between I and J
d_{ij}	:	distance between an antenna i (with coordinates (x_i, y_i)) and an edge data center j (with coordinates (x_j, y_j))
c_i	:	total number of CPU cores requested for processing the aggregated demands of antenna i
C_j	:	available computing resources (CPU cores) in each edge data center j
l_i	:	expected latency for processing the aggregated demands of antenna i
L_{ij}	:	transmission delay (latency) on the communication link between an antenna i and an edge data center j
x_{ij}	:	binary decision variable $$x_{ij} = \begin{cases} 1, & \text{if the antenna } i \in I \text{ is assigned} \\ & \quad \text{to the edge data center } j \in J \\ 0, & \text{otherwise} \end{cases}$$
y_j	:	binary decision variable $$y_j = \begin{cases} 1, & \text{if the edge data center } j \text{ is used} \\ 0, & \text{otherwise} \end{cases}$$

Before investigating new algorithms to deal with the constrained resource allocation problem, we address in the following the problem's complexity.

3.2.2 Problem complexity

We provide a theorem and a proof confirming the NP-Hardness of the RRH-BBU assignment problem.

Theorem 3.2.1 *Finding the optimal assignment of the antennas (RRHs) demands to the available edge data centers (BBU pools) is an NP-Hard problem.*

Proof As it is described above, the constrained resource allocation problem consists in finding the optimal assignment of antennas demands to the available edge data centers with the aim of satisfying latency and processing requirements and minimizing network resource utilization. Our problem is close to the Generalized Assignment Problem (GAP) (see [87] for more details), which is a classical generalization of both multiple knapsack problem [88] and bin packing problem [89]. Indeed, GAP consists in finding a feasible packing of the items (each item is defined by a size and a profit) into the bins (each bin has a limited capacity) that maximizes the total profit.

Our constrained resource allocation problem is very similar to GAP in which the antennas can be considered as items and edge data centers are the bins. Furthermore, compared to GAP, our constrained resource allocation problem has additional constraints concerning the latency requirements on the communication links joining the antennas and edge data centers. Hence, the relaxation of these constraints give an instance of GAP which means that the **optimal** solution of GAP is a **feasible** (not necessarily optimal) solution for RRH-BBU assignment problem.

Authors in [88] and [90] have proven the NP-Hardness of GAP. Therefore, by using the previous linear reduction from our problem to GAP, we deduce that our RRH-BBU assignment problem is also NP-Hard which means that finding the optimal assignment of the antennas demands to the available edge data centers is an NP-Hard problem.

■

The aim of the RRH-BBU assignment problem is to find an optimal mapping of all antennas demands on the available edge data centers when satisfying the processing and latency requirements. To achieve this objective, we discuss, in the following, an exact approach and heuristic algorithms to attend optimal and near-optimal solutions, respectively.

3.3 Proposed algorithms

In this section, we provide an exact approach based on ILP formulation to determine the optimal assignment of antennas demands to the edge data centers. Since our addressed problem is NP-Hard, we propose three approximation algorithms to efficiently deal with large instances of RRH-BBU assignment problem. It is worth noting that an ILP formulation is proposed to provide optimal (best) solutions for the RRH-BBU assignment problem for small and medium size networks and will be used to benchmark the performance of our proposed heuristic algorithms using several metrics.

3.3.1 Integer linear programming formulation

In this section, we formulate the RRH-BBU assignment problem by a mathematical formulation based on ILP approach. This allows to find optimal solutions to deal with small and medium problem instances. The decision variables and all parameters, that we will use in the mathematical formulation, are defined in Table 3.1.

Objective function

The objective of our constrained resource allocation problem is to efficiently assign the antennas demands to the most appropriate ("best") edge data centers when jointly satisfying the processing and latency requirements. This objective will be reached by finding the best trade-off between transport requirements on the fronthaul network and the number of active edge data centers. In fact, similarly to [56], [65], [82] and [91], our objective function in (3.1) contains two terms : the first denotes the total assignment cost in terms of transmission delay on the fronthaul network and the second term represents the total network resource utilization expressed by the number of used edge data centers. This is equivalent to select, in the final graph (as shown in the example of Figure 3.2), the optimal matching between the set of antennas and the available edge data centers when jointly optimizing the latency on the fronthaul network and the computing resource consumption.

$$\min \quad \mathcal{F} \quad = \sum_{j \in J} \sum_{i \in I} L_{ij} \times x_{ij} \quad + \quad \sum_{j \in J} y_j \tag{3.1}$$

Our constrained resource allocation problem has to comply with a number of constraints which will be summarized and mathematically expressed in the following.

Constraints

Constraints (3.2) guarantee that each antenna i should be connected **to exactly one** edge data center j. These constraints are considered in the graph solution of Figure 3.2 where each antenna is mapped on exactly one edge data center.

$$\sum_{j \in J} x_{ij} = 1, \quad \forall i \in I \tag{3.2}$$

Constraints (3.3) ensure that the assignment of antennas demands to the BBU pools does not violate the edge data centers' limited capacity constraints. In fact, as mentioned in Section 3.2.1, each edge data center j has a limited processing capacity C_j in terms of CPU cores and thus the total number of CPU cores requested for processing all antennas must not exceed the available computing resources of the selected edge data center.

$$\sum_{i \in I} c_i \times x_{ij} \leq C_j \times y_j, \quad \forall j \in J \tag{3.3}$$

Our optimization will select the most appropriate fronthaul links that satisfy the strict latency requirements of the antennas demands. In fact, constraints (3.4) impose that the transmission delay on the selected communication link between the antenna and the edge data center must not exceed the expected latency. Thus, as shown in Figure 3.2, only expected latencies will be kept in the final solution. This is guaranteed by the following inequalities:

$$L_{ij} \times x_{ij} \leq l_i, \quad \forall i \in I, \forall j \in J \tag{3.4}$$

Constraints (3.5) ensure that if there exists at least one antenna assigned to the edge data center j (i.e. $\sum_{i \in I} x_{ij} \geq 1$), then this edge data center is activated (i.e. $y_j = 1$) and can be used to host other antennas as long as its processing capacity is not exceeded. We recall that the optimal assignment of antennas demands to the edge data centers is reached when the number of used edge data centers is minimized. This will help network operators to reduce their network costs including CAPEX and OPEX.

$$y_j \leq \sum_{i \in I} x_{ij}, \quad \forall j \in J \tag{3.5}$$

Complete mathematical formulation

Our mathematical model is hence characterized by the following ILP formulation (3.6). Using an exact method, i.e. Branch-and-Bound (see Section 2.2.1.1 of Chapter 2), the proposed mathematical formulation explores all feasible solutions for the RRH-BBU assignment problem and selects the best one allowing to find the optimal strategies to assign the limited processing resources in the available edge data centers to the antennas demands. The solution provided by the ILP formulation (3.6) is **optimum** ("best" solution) and thus resource utilization gains can be achieved when the number of used edge data centers and the fronthaul latency are minimized.

$$\begin{aligned}
&\min \quad \mathcal{F} \quad = \textstyle\sum_{j \in J} \sum_{i \in I} L_{ij} \times x_{ij} \quad + \quad \sum_{j \in J} y_j \\
&S.T.: \\
&\textstyle\sum_{j \in J} x_{ij} = 1, \quad \forall i \in I \\
&\textstyle\sum_{i \in I} c_i \times x_{ij} \leq C_j \times y_j, \quad \forall j \in J \\
&L_{ij} \times x_{ij} \leq l_i, \quad \forall i \in I, \forall j \in J \\
&y_j \leq \textstyle\sum_{i \in I} x_{ij}, \quad \forall j \in J \\
&x_{ij}, y_j \in \{0, 1\}, \forall i \in I, \forall j \in J;
\end{aligned} \tag{3.6}$$

Since the NP-Hardness of our constrained resource allocation problem (see the proof in Section 3.2.2), the necessary convergence time to obtain optimal solutions

using an exact approach based on ILP formulation exponentially increases with the increase of number of antennas demands. Thus, we need to investigate new approximation algorithms that converge rapidly and provide optimal or near-optimal solutions for large problem instances. In the following, we introduce three heuristic algorithms (i) matroid-based approach, (ii) b-matching formulation and (iii) multiple knapsack-based algorithm. We recall that the obtained solution by the exact approach based on ILP formulation (3.6) is **optimum** ("best" solution) and will be used to evaluate the quality of solutions provided by the proposed heuristic algorithms.

3.3.2 Matroid-based approach

In addition to the exact approach based on the above description of the convex hull of the addressed problem, we seek a polynomial time algorithm that can scale to larger number of antennas and edge data centers. Since the optimal solution provided by ILP formulation in (3.6) is efficiently optimizing the latency and the resource allocation jointly, we propose new algorithm based on matroid theory [40] with similar properties and criteria.

Using the weighted bipartite graph $G = (I \cup J, E)$ constructed according to Figure 3.1b, the optimal solution of the constrained resource allocation problem consists in hosting each antenna demand in exactly one edge data center. Similarly, in the bipartite graph G, each vertex $i \in I$ will be assigned to exactly one vertex $j \in J$, and each vertex $j \in J$ can be a neighbor of different vertices in I as each edge data center can host more than one antenna. This yields a solution as presented in Figure 3.2b, showing a forest of trees optimally linking antennas I and edge data centers J. Thus, we propose the following theorem that defines our matroid for RRH-BBU assignment problem. We note that this matroid is well known in the literature and it is noted by the graphic matroid (see [25] for instance).

Theorem 3.3.1 *Let $G = (I \cup J, E)$ be a simple bipartite graph as shown in Figure 3.1b. By relaxing data centers' limited capacities constraints, $M = (E, \mathcal{F})$ is a matroid, with $\mathcal{F} = \{A \subseteq E, \quad A \ is \ a \ forest \ of \ trees\}$.*

In the following, we provide the proof of theorem 3.3.1 (we will use some concepts and definitions which are detailed in Section 2.2.1.3 of Chapter 2).

Proof Based on the bipartite graph $G = (I \cup J, E)$, we investigate trees decomposition with a minimum cost. In other words, and after the relaxation of data centers' limited capacity constraints, we seek to find an optimal basis of the graphic matroid.
In this proof, we use the definition of matroid that we introduced in Chapter 2 (Section 2.2.1.3). Thus, the proof is given as follows:

- The first condition (P1) of the definition concerning matroids, is trivial.

- The second condition (P2) of the matroid definition : Suppose we have $A \in \mathcal{F}$, and according to the definition of $\mathcal{F}$, A is a forest of trees. Thus, if $B \subseteq A$, then the connected components of B are also trees even by deleting one or multiple edges in A. This leads to easily conclude that $B \in \mathcal{F}$.

- To prove the last condition (P3) of the matroid definition, we note by $A = \cup_{i=1}^{k} A_i$ which represents the connected components (trees) of A. Then, for all $i = 1, \ldots, k$, we suppose $G_i = (T_i, A_i)$, where G_i is a tree with $|T_i|$ vertices and $|A_i|$ edges. This leads to deduce the number of vertices of A given by

$$n_A = \sum_{i=1}^{k} \mid T_i \mid = \mid A \mid + k. \tag{3.7}$$

  We also suppose $B = \cup_{j=1}^{t} B_j$, we note by $G_i^{'} = (T_i^{'}, B_i)$, where $G_i^{'}$ is a tree with $|T_i^{'}|$ vertices and $|B_i|$ edges. The number of nodes of B is then given by :

$$n_B = \sum_{j=1}^{t} \mid T_i^{'} \mid = \mid B \mid + t. \tag{3.8}$$

  By using $\mid B \mid > \mid A \mid$, two cases are discussed:

  1. If $n_B > n_A$ $(t > k)$: We suppose that B reaches more vertices than A, so there exists a vertex x covered by B and not by A. Suppose that $e \in B$ is an edge which contains x as one of its two extremities, we finally deduce that $A \cup \{e\} \in \mathcal{F}$.
  2. If $n_B < n_A$: We suppose that the edges of B connects each couple of nodes in A in the same connected component (tree) A_i. Using the absurd reasoning, we suppose that there is no edge $e \in B \setminus A$, leading to get $A \cup \{e\} \in \mathcal{F}$. This means that:
     - The edge $e \in B$, relies two vertices in the same component (tree) A_i and forms a cycle.

     In this case, the number of edges of B will verify $\mid B \mid \leq \mid V_1 \mid + \mid V_2 \mid + \ldots + \mid V_k \mid$, then $\mid B \mid \leq \mid A \mid$ which contradicts our hypothesis $\mid B \mid > \mid A \mid$.

■

As mentioned in Section 2.2.1.3 of Chapter 2, the above matroid formulation, defined by theorem 3.3.1, can be optimally solved by a simple greedy algorithm. However, this algorithm does not consider the constraints of limited processing capacity in the edge data centers. In fact, these constraints are strong in our RRH-BBU assignment problem because they influence the choice of which edge data center, the antenna will be assigned. Hence, we introduce some modifications in the matroid formulation to consider the edge data centers' limited capacity constraints. The complete matroid-based algorithm is illustrated in Algorithm 3.

Algorithm 3 Matroid-based algorithm for RRH-BBU assignment problem

```
Put A = ∅;
l_{e_1} ≤ l_{e_2} ≤ ... ≤ l_{e_m};
for i = 1 to m do
  if A ∪ {e_i} ∈ F then
    if c_{I(e_i)} ≤ C_{T(e_i)} then
      A := A ∪ {e_i}
      C_{T(e_i)} − = c_{I(e_i)}
    end if
  end if
end for
```

l_{e_i} is the communication latency on the edge e_i;
$I(e_i)$ (resp. $T(e_i)$) represents the initial (resp. terminal) extremity of the edge e_i;
$c_{I(e_i)}$ represents the number of CPU cores requested for processing the antenna demand $I(e_i)$;
$C_{T(e_i)}$ represents the available amount of CPU in an edge data center $T(e_i)$.

Matroid-based algorithm's complexity

It is important to evaluate the complexity of our proposed matroid-based algorithm. We note that the addressed problem is NP-Hard, and we need rapid and cost-efficient approaches to cope with this complexity. In fact, our proposed matroid-based algorithm has a global complexity (in the worst case) of $O\left(m\ln(m)+m\right)$, where the first term $m\ln(m)$ is the complexity of sorting a set of m edges according to their weights (latency in our case), and the second term m is the number of times, the "**For**" loop indicated in Algorithm 3 has been executed.

In addition to the matroid-based algorithm, we introduce in the following another heuristic algorithm based on b-matching approach. This proposal aims to find the optimal mapping between RRHs and BBUs, when satisfying all antennas demands. Using b-matching algorithm, we seek to rapidly reach optimal or near-optimal solutions for large instances of RRH-BBU assignment problem. This may not be feasible with matroid-based approach, especially when the number of antennas demands becomes important (more than 100 antennas) and the computing resources in available edge data centers are limited.

3.3.3 b-Matching formulation

In this section, we propose a new heuristic approach based on b-matching theory [39] to address larger problem instances and to attend optimal or near optimal solutions in negligible times. The b-matching algorithm uses the bipartite graph, as described in Figure 3.1b, to find the minimum weight b-matching between antennas and edge data centers when jointly reducing the number of used edge data centers and the latency requested for processing the aggregated demands.

According to the definition of b-matching approach (see Section 2.2.1.3 of Chapter 2), we introduce new algorithm that solves the constrained resource allocation

problem by finding the minimum weight b-matching between antennas and edge data centers in the bipartite graph $G = (I \cup J, E)$. This algorithm will jointly consider the strong latency requirements of antennas demands and the limited processing capacity constraints of the edge data centers.

Proposition 3.3.2 *Let $G = (I \cup J, E)$ be a weighted bipartite graph. The constrained resource allocation problem defined above can be solved by finding the minimum weight b-matching while considering the following parameters:*

- *The integral edge capacities : $u = 1$.*
- $b(i) = 1, \quad \forall i \in I$ *(I is a set of antennas).*
- $b(j) = min\{|I_j|, \lfloor \frac{C_j}{\overline{c(j)}} \rfloor\}, \quad \forall j \in J$ *(J is a set of edge data centers).*

where :

- I_j is a subset of antennas that can be assigned to the edge data center $j \in J$ when satisfying the expected latency and CPU cores amount requested for each antenna demand : $I_j = \{i \in I \quad | \quad (l_i \geq L_{ij}) \quad \wedge \quad (c_i \leq C_j)\}$.
- $\overline{c(j)}$ is the average number of CPU cores of antennas demands that can be assigned to the edge data center $j \in J$: $\overline{c(j)} = \frac{\sum_{i \in I_j} c_i}{|I_j|}$.

In addition and in order to help our optimization to find optimal solution with integer variables, we add the blossom inequalities given by the following formula (3.9).

$$\sum_{e \in E(G[X])} x_e + \sum_{e \in F} x_e \leq \lfloor \frac{1}{2} (\sum_{v \in X} b(v) + |F|) \rfloor, \quad \forall X \subseteq I \cup J, F \subseteq \delta(X) \tag{3.9}$$

where $E(G(X))$ represents a subset of edges in the subgraph $G(X)$ generated by a subset of vertices X and $\delta(X)$ is a set of incident edges of X (more details can be found in Section 2.2.1.3 of Chapter 2).

Finally, we use the obtained result of Proposition 3.3.2 to define a new minimum weighted b-matching formulation to polynomially solve the constrained resource allocation problem. The mathematical formulation is given by the following model:

$$\min \quad \mathcal{F} \quad = \sum_{e \in E} L_e \times x_e$$

$$\begin{aligned}
&S.T. : \\
&\textstyle\sum_{e \in \delta(i)} x_e = 1, \quad \forall i \in I; \\
&\textstyle\sum_{e \in \delta(j)} x_e \leq min\{|I_j|, \lfloor \frac{C_j}{\overline{c(j)}} \rfloor\}, \quad \forall j \in J; \\
&\textstyle\sum_{e \in E(G[X])} x_e + \sum_{e \in F} x_e \leq \lfloor \frac{1}{2} (\sum_{v \in X} b(v) + |F|) \rfloor, \quad \forall X \subseteq I \cup J, F \subseteq \delta(X); \\
&x_e \in \mathbb{R}^+, \quad \forall e \in E;
\end{aligned} \tag{3.10}$$

b-Matching algorithm's complexity

To assess the ability of our b-matching algorithm to find good solutions for large-scale graph instances in reasonable times, we analyze in this section the complexity of the proposed algorithm. We note that the objective of this algorithm is to rapidly assign antennas demands to available edge data centers under strict latency requirements and limited processing capacity constraints. The complexity of our proposed b-matching approach based on linear programming is $O\left(|V|^2|E|\ln(\frac{|V|^2}{|E|})\right)$ where $V = I \cup J$ and E is the set of weighted links between I and J. This approach is a simple linear program with a negligible complexity. For interested readers, more details can be found in [43] and [44].

In the following, we provide another heuristic algorithm using the multiple knapsack formulation. The multiple knapsack approach has been very well used in the literature (see for instance [27], [92] and [93]) to address resource allocation problems in different contexts. In our context of C-RAN, we propose a heuristic algorithm based on multiple knapsack formulation to solve the RRH-BBU assignment problem. The obtained solutions by this algorithm will be benchmarked with matroid and b-matching algorithms to better evaluate the performance of our algorithms using different metrics.

3.3.4 Multiple knapsack-based approach

In addition to the exact mathematical formulation and heuristic algorithms proposed above, we propose a new algorithm using the well known multiple knapsack formulation to address the RRH-BBU assignment problem when considering the limited capacities of edge data centers. In fact, the multiple knapsack formulation is a generalization of the classical knapsack problem from a single knapsack to m knapsacks with different capacities. The objective of multiple knapsack algorithm is to assign each item to at most one of the knapsacks such that none of the capacity constraints are violated and the total profit of the items put into knapsacks is maximized.

According to the definition of multiple knapsack formulation (see Section 2.2.1.3 of Chapter 2) and by considering the bipartite graph $G = (I \cup J, E)$, we obtain the following equivalence between our constrained resource allocation problem and the multiple knapsack formulation :

- The knapsacks are the edge data centers ($j \in J$).
- The antennas demands ($i \in I$) are the items to be inserted in the knapsacks (data centers).
- The weight w_j is the amount of CPU cores c_i requested for processing the antenna demand i.
- The profit p_j does not vary between different antennas demands and can be set to 1 ($p_j = 1$).

The previous formulation addresses the constrained resource allocation problem by only focusing on the processing capacity of the edge data centers when relaxing the latency requirements of antennas demands. This relaxation influences the choice of which edge data center will host the antennas demands. Hence, in order to consider these constraints in the final solution, we introduce a simple modification in the multiple knapsack algorithm which consists in checking if the expected latency is guaranteed before assigning the antenna demand to the edge data center. We illustrate our multiple knapsack formulation in Algorithm 4.

Algorithm 4 Modified Multiple Knapsack Algorithm

Input: $G = (I \cup J, E)$, Antenna demands, Edge data centers.
Output: A joint mapping (CPU, Latency) of all antennas demands on the available edge data centers.
This is summarized formally in steps:

Step 1: Sort the edge data centers ($j \in J$) in increasing order of their CPU capacities C_j;

Step 2: Select the antennas demands that can be assigned to the selected edge data center j by checking if :

- The expected latency of the antenna demand is provided by the communication link joining it to the selected edge data center j;
- The available computing resources in the selected edge data center j are greater than the number of CPU cores requested by the antenna demand;

Step 3: Pick as many antennas demands as possible to the selected edge data center using the dynamic programming approach (see Algorithm 2);
Step 4: Update the total number of available CPU cores in the selected edge data center;
Step 5: Repeat Steps 2, 3 and 4 until all considered antennas demands are assigned to the edge data centers;

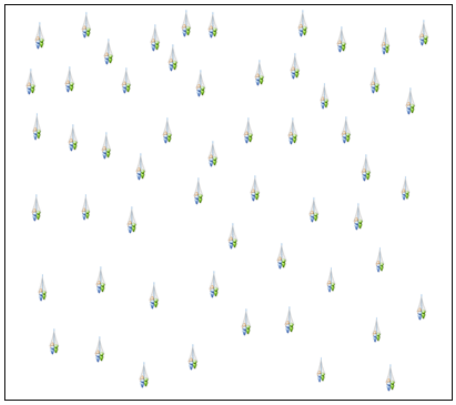
3.4 Performance evaluation

The simulation and experiments use the optimization solver Cplex [94] for the linear programming approaches, the exact approach based on ILP formulation in (3.6) and the b-matching formulation in (3.10). We evaluate the performance of the exact algorithm and then we compare the obtained solutions (optimum) with those found by our heuristic algorithms in terms of convergence time, scalability and optimality. Each simulation scenario is run 100 times using different parameters.

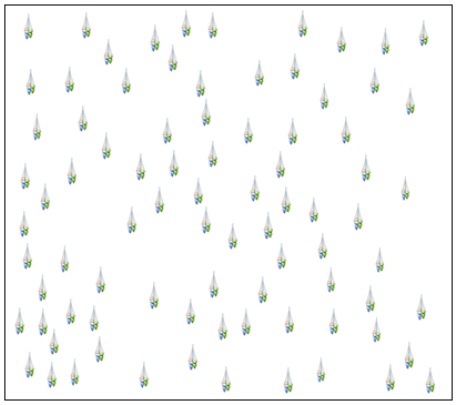
3.4.1 Simulation settings and parameters

The performance evaluation of our algorithms is conducted using a 2.40 GHz PC with 8 GB RAM. The number of antennas is generated following a Poisson process with a parameter $\Lambda = \lambda \times space_dimensions$, where λ is varying in the range $[0.1; 1]$, and *space_dimensions* in the range $[5; 20]$. In Figure 3.3, we illustrate four examples of simulation scenarios when considering a cellular network in a region of space dimensions $space_dimensions = 10 \times 10$ and varying the density of antennas $\lambda \in \{0.3; 0.5; 0.8; 1\}$.

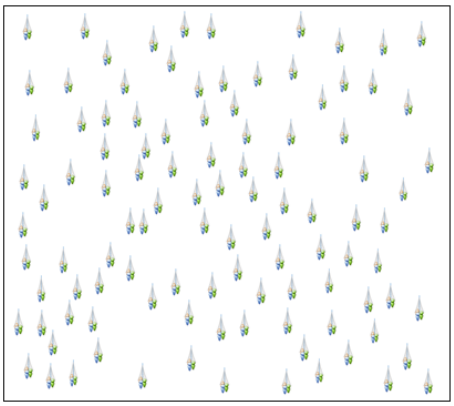
(a) $space_dimensions = 10 \times 10$, $\lambda = 0.3$



(b) $space_dimensions = 10 \times 10$, $\lambda = 0.5$



(c) $space_dimensions = 10 \times 10$, $\lambda = 0.8$



(d) $space_dimensions = 10 \times 10$, $\lambda = 1.0$

Figure 3.3: Example of simulation scenarios for RRH-BBU assignment problem

Each antenna comprises a random number of demands (from end-users) presented in terms of an amount of CPU cores in the $[5; 10]$ interval (some papers such as [60] and [95] are considering allocation of Physical Resources Blocks PRBs, this is not changing our mathematical modeling and the convergence of our algorithms to good solutions). The number of edge data centers is set to 20 each of which has random computing resources or number of available CPUs drawn in the $[50; 200]$

CPU cores range. The workloads (i.e. aggregated amount of end-users demands in terms of equivalent CPU cores) of the antennas demands are expecting a latency to not exceed 1 millisecond and this is drawn randomly in the $[0.1; 1]$ milliseconds range. For sake of clarity, we summarize the simulation settings and parameters in Table 3.2.

Table 3.2: RRH-BBU assignment algorithms : simulation settings and parameters

Parameters	**Values**
Density of antennas	$\lambda \in [0.1; 1]$
Space dimensions	10×10; 20×20;...
Poisson parameter	$\Lambda = \lambda \times space_dimensions$
Number of antennas	Poisson distribution: $\mathcal{P}(\Lambda)$
Antenna coordinates	Uniform distribution: $\mathcal{U}(0, space_dimensions)$
Number of edge data centers	20
Latency between antenna i and edge data center j	5 µs/km
Expected latency of antenna i	$l_i \in [0.1ms;\ 1ms]$
Number of CPU cores required by each antenna i	$c_i \in [5; 10]$
Number of CPU cores in each edge data center j	$C_j \in [50; 200]$

3.4.2 Performance metrics

The metrics used for the performance assessment of our algorithms (exact and heuristics) are detailed in the following :

- **Convergence time**: is the time needed by the algorithms to converge to their best solutions.

- **Resource utilization rate**: is defined as the percentage of edge data centers that are used to host the aggregated antennas demands and it can be expressed as follows :

$$Resource\ utilization\ rate(\%) = \frac{\sum_{j \in J} y_j}{|J|} \times 100 \tag{3.11}$$

  where $|J|$ is the total number of available edge data centers.

- **Gap**: is used to benchmark the proposed heuristics with the exact ILP algorithm used as "reference and optimal solution". With no loss of generality, we focus on the comparison of CPU resource consumption (expressed by the percentage of edge data centers used to host all antennas demands). We note

that the quality of the solution provided by the heuristic algorithms is better when the cost gap value is smaller (**optimum when the gap is equal to 0**). This metric is formally expressed as:

$$Gap(\%) = \mid Utilization\ rate(\mathbf{ILP}) - Utilization\ rate(\mathbf{Heuristic}) \mid \quad (3.12)$$

- **Rejection rate**: is the average of the percentage of antennas demands that cannot be assigned to each edge data center. This metric, can be expressed as a function of the decision variables and parameters described in Table 3.1 :

$$Rejection\ rate(\%) = \frac{|I| - \sum_{j \in J} \sum_{i \in I} x_{ij}}{|I|} \times 100 \quad (3.13)$$

  where $|I|$ is the total number of antennas.

- **SLA violations rate**: is the average of over-used edge data centers in terms of CPU cores. This metric will be mainly used to evaluate the ability of the matroid-based approach in finding optimal solutions that **do not violate** the edge data centers' limited capacity constraints (which is defined, in the ILP formulation, by constraints (3.3)). We only focus on matroid-based algorithm (as defined by theorem 3.3.1) because there are no SLA violations with ILP, b-matching and multiple knapsack appraoches. The average of SLA violations rate can be expressed as a function of decision variables and parameters (described in Table 3.1).

$$SLA\ violations\ rate(\%) = \frac{1}{|J|} \times \sum_{j \in J} \frac{\sum_{i \in I} c_i \times x_{ij} - C_j \times y_j}{C_j \times y_j} \times 100 \quad (3.14)$$

  where $|J|$ is the total number of available edge data centers.

3.4.3 Performance analysis

3.4.3.1 Performance evaluation of ILP based approach

Table 3.3 depicts the performance results in terms of convergence time and rejection rate of the exact algorithm based on ILP formulation. This algorithm explores all feasible solutions before finding the optimum. This causes an exponential increase of the convergence time when increasing the number of antennas. Indeed, the ILP approach needs more than 4 minutes (4.39 minutes) to converge to optimal solutions for an instance of 400 antennas and 20 available edge data centers. This is expected since the addressed problem is NP-Hard. Thus, the ILP approach can be used for small or medium instances with a number of antennas not exceeding 100. Furthermore, the rejection rate is always equal to 0 which means that the exact approach based on ILP formulation is always able to assign all antennas demands to the available edge data centers.

Table 3.3: Performance of the exact approach based on ILP formulation

Space	λ	**#Antennas**	**Convergence time**	**Rejection rate**
10×10	0.3	30	**9.63s**	0
	0.5	50	**10.92s**	0
	0.8	80	**11.87s**	0
	1	100	**12.58s**	0
20×20	0.3	120	**62.09s**	0
	0.5	200	**86.56s**	0
	0.8	320	**2.87min**	0
	1	400	**4.39min**	0

3.4.3.2 Performance evaluation of heuristic algorithms

In Table 3.4, we consider different simulation scenarios by varying the dimensions of the considered space area as well as the density of deployed antennas (see the examples in Figure 3.3). Using these simulations, we would like to evaluate the performance of our proposed approximation algorithms: matroid-based algorithm (Algorithm 3), b-matching formulation given by (3.10) and the Multiple knapsack-based approach (Algorithm 4).

Table 3.4: Heuristic algorithms' performance assessment

Space	λ	**#Antennas**	**Heuristic algorithm**	**Convergence time**	**Gap(%)**	**Rejection rate(%)**
10×10	**0.3**	**30**	matroid	0.28**ms**	7	0
			b-matching	0.34s	7	0
			multiple knapsack	0.57**ms**	8	0
	0.5	**50**	matroid	0.38**ms**	5	0
			b-matching	0.36s	5	0
			multiple knapsack	1.01**ms**	11	0
	0.8	**80**	matroid	0.51**ms**	6	0
			b-matching	0.26s	5	0
			multiple knapsack	1.69**ms**	15	0
	1	**100**	matroid	0.88**ms**	6	0
			b-matching	0.39s	4	0
			multiple knapsack	3.35**ms**	15	0
20×20	**0.3**	**120**	matroid	0.94**ms**	-	1
			b-matching	0.4s	4	0
			multiple knapsack	4.35**ms**	-	1
	0.5	**200**	matroid	1.02**ms**	-	4
			b-matching	0.39s	6	0
			multiple knapsack	7.71**ms**	-	1
	0.8	**320**	matroid	1.75**ms**	-	17
			b-matching	0.34s	6	0
			multiple knapsack	25.44	-	3
	1	**400**	matroid	2**ms**	-	19
			b-matching	0.33s	5	0
			multiple knapsack	39.89**ms**	-	4

As shown in Table 3.4, our heuristic algorithms are benchmarked with the ILP approach, that provides optimum solutions, using three performance metrics : the convergence time, the gap (3.12) to compare with optimal solutions provided by the exact approach and the rejection rate (3.13). We note that we calculate the gap only if the rejection rate is equal to 0, otherwise it is not really significant.

Table 3.4 highlights clearly the efficiency of the matroid-based algorithm in finding near optimal solutions faster than the exact approach based on ILP formulation. Indeed, the matroid approach provides good solutions with an average gap not exceeding 7% in worst cases and needs 2 **milliseconds** to converge when considering large graphs of 400 antennas and 20 available edge data centers. Thus, the matroid-based approach can be used to cope with large problem instances. However, the matroid approach comes with some drawbacks such as it cannot assign all antennas demands for large problem instances. This is shown by the rejection rate metric of which its value can reach 19% for an instance of 400 antennas and 20 available edge data centers.

To better evaluate the performance of our matroid-based algorithm, we calculate the rejection rate when increasing the number of considered edge data centers. For that, we consider two network instances of 320 and 400 antennas and we varied the number of edge data centers from 20 to 60. The obtained results of these simulations are represented by Figure 3.4.

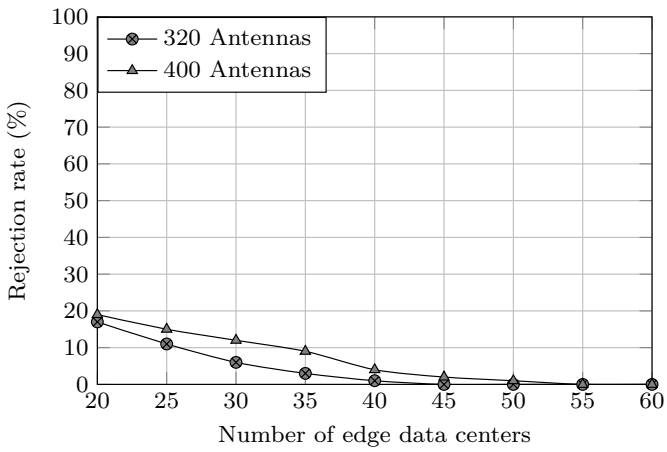


Figure 3.4: Matroid-based approach : rejection rate variation when increasing number of edge data centers

The simulation results in Figure 3.4 show that the rejection rate depends on the amount of available computing resources and thus decreases when the number of available edge data centers increases. In fact, for the first simulation scenario (320 antennas), matroid-based algorithm attends a rejection rate equal to 0 when there are at least 40 available edge data centers, while for the second simulation scenario (400 antennas), the rejection rate vanishes when there are at least 50 available edge data centers. This means that the matroid-based algorithm becomes more efficient when more resources (edge data centers) are considered.

In addition and in order to get a better grasp of the relative performance of the

matroid-based approach, we illustrate in Figure 3.5 the SLA violations rate behavior according to different network sizes. In fact, we consider 4 simulation scenarios : $50, 100, 200, 320$ antennas to be efficiently assigned to a number of edge data centers ranging from 20 to 100. We recall that, for this simulation, we consider the matroid-based algorithm (as defined in theorem 3.3.1) when relaxing the edge data centers' limited capacity constraints and we calculate the SLA violations rate as defined by Formula (3.14).

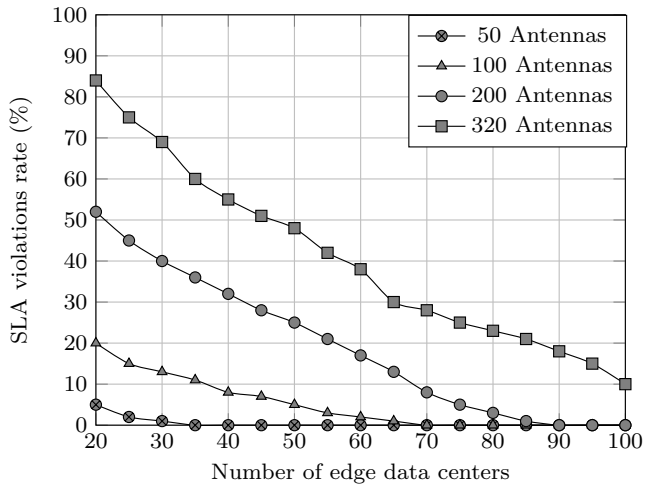


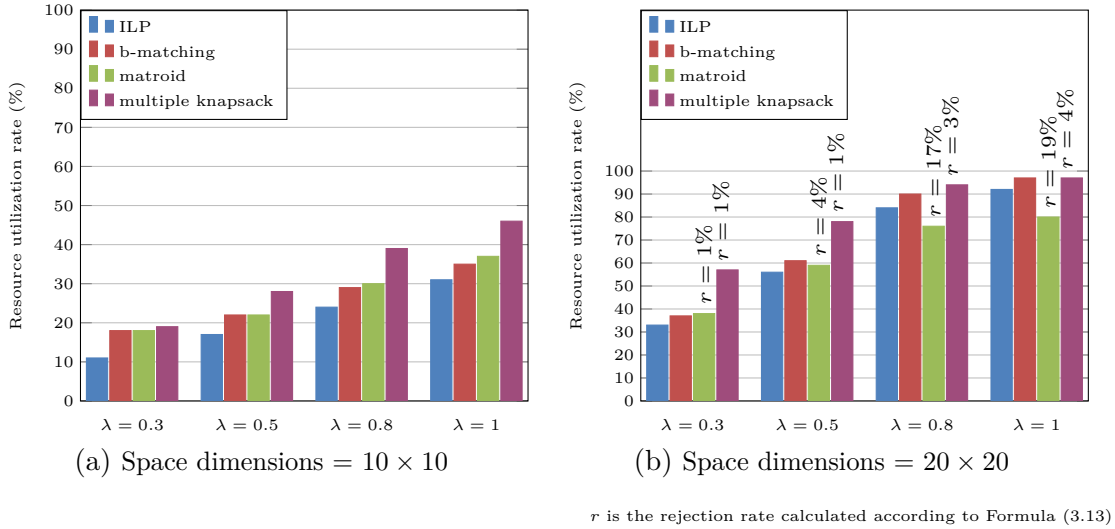
Figure 3.5: SLA violations rate behavior of the matroid-based approach

Simulation results in Figure 3.5 confirm that the SLA violations rate decreases when more processing resources (edge data centers) are considered. This confirms that the efficiency of the matroid-based algorithm depends on the amount of the available processing resources and attends good solutions when more resources (edge data centers) are used.

3.4.3.3 Resource utilization behavior

Figure 3.6 depicts the percentage of resource utilization (in terms of number of used edge data centers) obtained by the exact approach based on ILP formulation, which provides optimal solutions, and by the three approximation algorithms (matroid, b-matching and multiple knapsack). We recall that the value of the resource utilization rate is calculated according to Formula (3.11). With a weak advantage of the ILP method which consists in investigating all the feasible solutions before keeping the optimal one, the matroid-based approach and b-matching algorithms can find an efficient assignment of antennas demands to the available edge data centers while the solution obtained by multiple knapsack algorithm consumes a larger number of edge data centers (as shown in Figure 3.6a).

It is important to mention that for larger problem instances (Figure 3.6b), b-matching algorithm always provides good solutions in terms of resource utilization, close to the optimum solution by ILP, with a **rejection rate equal to 0**%. However, for matroid and multiple knapsack algorithms, the resource utilization rate



(a) Space dimensions = 10 × 10

(b) Space dimensions = 20 × 20

Figure 3.6: Resource utilization in different space dimensions

depends on the rejection rate (negligible but different from zero) in the case of large network size (see the legends appended to the resource utilization rate for matroid and multiple knapsack algorithms in Figure 3.6b). Therefore, we deduce that **b-matching algorithm can easily scale when large problem instances are considered** and thus can be used by network operators to efficiently reduce their network costs (CAPEX and OPEX) and achieve network utilization gains.

3.4.3.4 Algorithm's performance evaluation using real traces

To better evaluate the performance of our proposed algorithms, we consider a real trace from a 4G-LTE cell map of the network operator Orange, in a small area in Paris [3]. As shown in Figure 3.7, this topology represents a cellular network containing 50 antennas with their given geographical positions (coordinates). Then, according to [11] and [17], we place 20 edge data centers on the cell map such that the distance separating the antennas and the edge data centers is limited between 20 and 40 Kilometers. Similarly to the simulation parameters described in Table 3.2, we consider that each edge data center have a limited capacity of processing in terms of CPU cores while the antennas demands have variable processing and latency requirements.

In this experimentation, we apply our exact approach based on ILP formulation (3.6) and the three proposed approximation algorithms, including matroid-based algorithm (Algorithm 3), b-matching formulation (3.10) and multiple knapsack-based approach (Algorithm 4), on the 4G-LTE cell map of Figure 3.7. The solutions provided by these algorithms are benchmarked according to three performance metrics : convergence time, resource utilization rate given by (3.11) and rejection rate defined by (3.13).

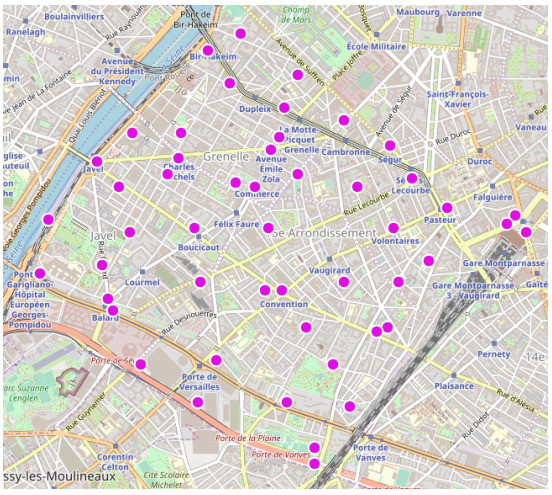


Figure 3.7: Real trace : Orange 4G-LTE cell map in Paris. Source: [3]

Table 3.5 shows that both matroid-based approach and b-matching formulation provide **optimal** solutions (the same solution provided by the ILP approach) in negligible times. In fact, with a weak advantage of the matroid-based approach which converges to the optimum in **23.52 ms**, the b-matching algorithm can also find an efficient assignment of antennas demands to the available edge data centers in **0.58 ms**. However, the solution obtained by multiple knapsack algorithm consumes a larger number of edge data centers, with a resource utilization rate equal to 25%. Regarding the rejection rate metric, all proposed algorithms can assign all considered antennas demands to the the available edge data centers and satisfy their latency and processing requirements without SLA violations.

Table 3.5: Performance evaluation using a real cellular network in Paris

Algorithm	**Convergence time (ms)**	**Resource utilization rate (%)**	**Rejection rate (%)**
ILP formulation	334.21	15	0
b-Matching algorithm	23.52	15	0
Matroid-based approach	0.58	15	0
Multiple knapsack algorithm	1.7	25	0

3.4.3.5 Scalability evaluation

The performance assessment would not be complete without addressing the scalability for very large problem instances. In fact, we propose a simulation scenario with an instance of **400** antennas and number of edge data centers in {**60**, **80**} which are both generated according to the parameters detailed in Table 3.2. Simulation results in Table 3.6 confirm the efficiency of matroid-based approach and

b-matching algorithm in finding good solutions in negligible times compared to ILP approach. Indeed, the matroid algorithm provides near optimal solutions (gap not exceeding 2%) in less than **28 milliseconds** and the b-matching algorithm can optimally solve the assignment problem in less than **4 seconds** (with gap value not exceeding 3%). However, the ILP approach is not converging in more than **1 hour** due to the exploration of all feasible solutions.

Regarding the multiple knapsack algorithm, the gap is a bit high compared to matroid and b-matching algorithms and reaches 19% for an instance of 400 antennas and 80 edge data centers.

Table 3.6: Algorithms' scalability assessment

#Antennas[1]	**#Edge[2]**	**ILP**	**b-matching**		**matroid**		**multiple knasapck**	
		Time	**Time**	**Gap**	**Time**	**Gap**	**Time**	**Gap**
400	60	**34.28min**	1.47s	3	6.42ms	2	82.84ms	18
	80	**1.02hour**	3.97s	2	27.16ms	2	107.4ms	19

This simulation is executed 100 times with different parameters.
[1] Antennas are generated as described in Table 3.2.
[2] Edge data centers.

3.4.3.6 Comparative analysis of proposed algorithms

In this section, we present a comprehensive comparison of the proposed algorithms for the constrained resource allocation (RRH-BBU assignment) problem. A taxonomy of these approaches in terms of: i) computational complexity ii) cost savings (including OPEX and CAPEX), iii) scalability, iv) implementation difficulty are highlighted in Table 3.7. As shown in this table, the matroid and b-matching algorithms are globally more efficient in finding good solutions in negligible times and in scaling larger problem instances. However, we note that it is not easy to implement the b-matching algorithm (described by mathematical formulation (3.10)) due to the complex implementation of the blossom inequalities (3.9).

Table 3.7: Algorithms' qualitative comparison

Algorithm	**Complexity**	**Cost savings**	**Scalability**	**Implementation difficulty**
ILP formulation	Exponential	■■■■	■□□□	■■□□
b-Matching algorithm	Polynomial	■■■□	■■■□	■■■□
Matroid-based algorithm	Logarithmic	■■■□	■■■■	■□□□
Multiple knaspack algorithm	Linear	■□□□	■■■□	■■□□

3.5 Conclusion

In this chapter, we addressed the constrained resource allocation problem (RRH-BBU assignment problem) with the objective of determining the best strategies to assign antennas demands to available edge data centers when jointly optimizing communication latency and resource consumption. Hence, we proposed an exact algorithm based on ILP formulation to find optimal solutions for small and medium size networks. The exact algorithm optimizes the resource consumption (in terms of active edge data centers) and communication latency associated for assigning antennas demands to the most appropriate edge data centers. However, this algorithm is known to not to scale for large problem instances. Therefore, we proposed three approximation algorithms based on exact theories and approaches : matroid-based approach, b-matching algorithm and multiple knapsack-based algorithm to meet larger number of antennas demands in negligible times.

The performance evaluation has been conducted using different simulation scenarios and five performance metrics. The simulation results have revealed the efficiency of the matroid-based approach and b-matching algorithm in terms of optimal solutions and convergence time even for large problem instances. This is confirmed by the numerical results when considering a real trace from a 4G-LTE cell map.

Nevertheless, the optimal assignment of antennas to the edge data centers, where many RRHs share common BBU computational resources, could be achieved when the inter-cell interference are reduced. Thus, in the next chapter, we will investigate new optimal approach to reduce interference between antennas when guaranteeing full network coverage.

Chapter 4

Full network coverage optimization in C-RAN

4.1 Introduction

The significant gain in terms of latency, resource utilization and cost savings, achieved by the optimal RRH-BBU assignment (detailed in the previous chapter), is largely constrained by the increasing of inter-cell interference that severely decreases the end-users' QoS. In fact, network operators are investigating new solutions to increase the density of existing cells by deploying more antennas in order to enlarge network spectrum and fulfill end-users requirements. Network densification is considered as a key method in C-RAN architecture to enhance network coverage and capacity and achieve higher data rates. However, this brings a variety of challenges including the management and reduction of inter-cell interference, generated by the dense deployment of antennas, and the optimization of network coverage by detecting and eliminating coverage holes. Consequently, it is crucial for network operators to investigate new approaches that enable to find a good tradeoff between inter-cell interference elimination/reduction and network coverage optimization.

In this chapter, we seek new approaches that enable to consolidate and re-optimize the antennas radii in order to reduce inter-cell interference when maintaining the full network coverage in C-RAN. We propose a new mathematical model based on ILP formulation to describe the convex hull of full network coverage optimization problem. Then, we enlarge this description by adding new valid inequalities and cutting planes to accelerate the convergence time and reach optimal solutions even for large number of antennas. Finally, we propose a deep Branch-and-Cut algorithm based on these cutting planes to efficiently solve the large-scale problem instances and we evaluate the efficiency and usefulness of our proposed approach compared to those proposed in the state of the art.

The rest of the chapter is organized as follows. In Section 4.2, we describe the problem statement and the network topology that will be used and then we discuss the complexity of our problem. In Section 4.3, we provide a Branch-and-Cut approach describing the convex hull of the joint interference and coverage optimization problem and we reinforce this formulation by proposing new families of valid

inequalities to accelerate the convergence time to the optimum. Numerical results and performance assessment can be found in Section 4.4 followed by a conclusion in Section 4.5.

4.2 Problem statement

In this section, we present the system model that we use to define the full network coverage problem in the context of C-RAN. Then, we provide a complexity study of the addressed problem.

4.2.1 System model and problem description

We consider a cellular network deployed in a large area represented by a set of antennas denoted by $\mathcal{A}$. Each antenna i is defined by its position on the plane and its coverage radius r_i which varies in the range $[r_i^{min}; r_i^{max}]$, where $i = 1, \ldots, |\mathcal{A}|$. The coverage area of each antenna is modeled by circles with variable coverage radius.

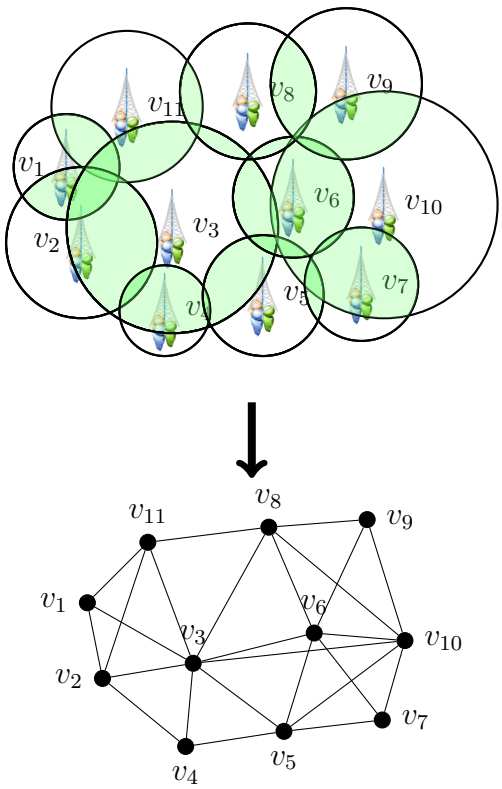


Figure 4.1: System model: graph construction based on antennas positions and interference

In fact, in real life, cells have random shape of coverage area which depends on geographic, environmental and network parameters (base station location, transmission power, terrain and artificial structures properties, ...). In the literature and for sake of representation and analytical simplicity, approximate approaches are often adopted to design and model the cells' coverage area in cellular networks. In particular, [96], [97] and [98] used hexagons to model the cells coverage area with no overlap between cells. This approximation is frequently employed in planning and analyzing wireless networks due to its flexibility and convenience. However, since the hexagons are only an idealization of the irregular cell shape, a simpler approach, called circular-cell approximation, is used to model the cell coverage area by circles (see [99], [100] and [101] for example). The circular-cell approximation is reasonable and very used in the modeling of cellular networks due to its low computational complexity. Hence, some references (see [102] and [103]) are using this approach to address the network coverage problem, and the authors proposed methods and algorithms that do not converge in acceptable times and do not provide good solutions. Our optimization is using circles to represent antennas coverage areas, and we propose an exact formulation that always provide optimal solutions in negligible times.

As depicted in Figure 4.1, we represent our network using an undirected graph denoted by $G = (\mathcal{A}, \mathcal{E})$ where $\mathcal{A}$ and $\mathcal{E}$ are the sets of available nodes and edges, respectively. There is an edge (i, j) between two antennas i and j if the following condition (4.1) is met :

$$r_i + r_j \geq d_{ij} \tag{4.1}$$

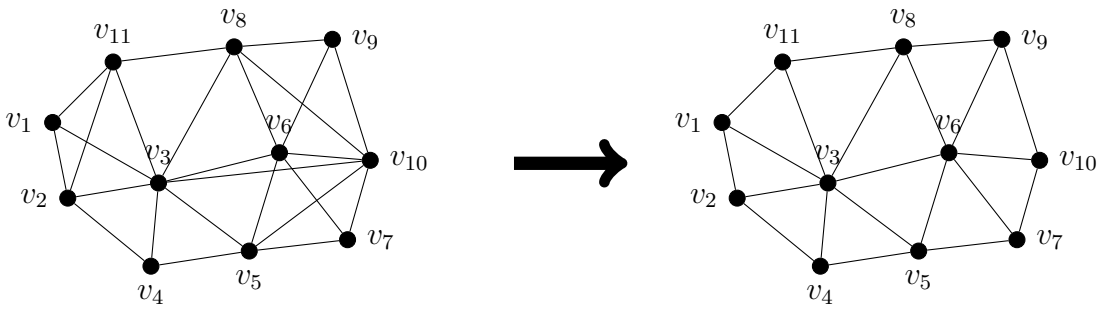
where r_i and r_j are the radii of antennas i and j respectively, and d_{ij} is the Euclidean distance between i (with the coordinates (x_i, y_i)) and j (with the coordinates (x_j, y_j)) and provided by:

$$d_{ij} = \sqrt{(x_j - x_i)^2 + (y_j - y_i)^2} \tag{4.2}$$

Let δ_{ij} be the overlapping (inter-cell interference) caused by two antennas i and j.

$$\delta_{ij} = r_i + r_j - d_{ij} \tag{4.3}$$

This overlapping is causing interference that should be reduced or totally eliminated when considering full network coverage and connectivity. Thus, similarly to the simplicial homology and Delaunay triangulation approaches (detailed in chapter 2), we aim to extract, from a given graph G (lower part of Figure 4.1), a subgraph G_Δ composed by adjacent triangles, each of which represents a complete coverage of the area around three antennas (according to the definition of Rips complex which can be found in Section 2.2.2 of Chapter 2). These graphs are illustrated in Figure 4.2. We note that the triangulated graph in the right part of Figure 4.2 represents a total covered network with minimum inter-cell interference.



(a) Initial Graph (b) Final Graph

Figure 4.2: A graph solution example of the full coverage network problem

4.2.2 Problem complexity

As mentioned before, our full network coverage problem consists in constructing a graph composed only by adjacent triangles. This triangulation method can be assimilated to the Minimum Weight Triangulation (MWT) problem which consists in finding in a graph G a set of edges of minimum total weight that triangulates the total nodes of G. This problem has been proven to be NP-Hard [104]. In the following, we discuss the complexity of the full network coverage problem when the reached optimum solutions **do not violate** any of the problem constraints that will be defined below.

Theorem 4.2.1 *For an instance of the optimal full network coverage problem defined above, deciding whether a solution with no violations exists is NP-Complete.*

Proof To prove this theorem, we will proceed according to the following steps:
It is important to recall that to cope efficiently with our problem, we investigated a polyhedral approach describing a set of valid inequalities to attend optimal solutions using a Branch-and-Cut strategy. This polyhedral approach will lead to find an optimal minimum weight triangulation when eliminating totally network interference represented by the set of edges intersections. This family of valid inequalities is illustrating the main difference between the full network coverage problem that we are addressing in our work and the MWT problem.

For sake of clarity, and for a given instance of a weighted graph $G = (\mathcal{A}, \mathcal{E})$, let φ^*_{mwt} be the optimal value of the MWT problem, and $\psi^*_{hole_cov}$ the optimum found when solving the full network coverage problem. As our problem is more constrained compared to the MWT problem, then we deduce that $\varphi^*_{mwt} \leq \psi^*_{hole_cov}$. This implies that, the relaxation of the constraints which consists in eliminating existing interference (edges intersections) will hold to retrieve an instance of the MWT problem. Indeed, the **optimal** solution of the minimum weight triangulation problem is a **feasible** (not necessarily optimal) solution in the full network coverage problem instance.

In addition, in 2006, W. Mulzer and G. Rote (see [104]) have proven the NP-Hardness of the MWT problem. Thus, by using the previous linear reduction from

our problem to the MWT problem, we conclude that the full network coverage problem is also NP-Hard. This implies that the decision formulation concerning the existence of no violation solutions of the full network coverage problem is NP-Complete.

■

Our problem is then NP-Complete, and we need rapid approaches to attend optimal solutions in acceptable times. Our proposal is based on the construction of a complete description of the convex hull of the incidence vectors characterizing the optimal solution of the full network coverage problem.

4.3 Branch-and-Cut formulation

To cope with the full network coverage problem, we propose a Branch-and-Cut algorithm based on the description of the convex hull of the problem's incidence vectors. This description consists in various families of valid inequalities leading to attend the optimal solution in acceptable times.

4.3.1 Convex hull characterization

Before introducing our mathematical formulation, we start by providing the variables and parameters that will be used our formulation.

- We consider our initial graph $G = (\mathcal{A}, \mathcal{E})$ representing the network topology as illustrated by the lower part of Figure 4.1. $\mathcal{A}$ is the set of antennas and $\mathcal{E}$ is the set of edges between antennas. According to formula (4.1), we populate the graph G (see lower part of Figure 4.1).
- Each antenna $i \in \mathcal{A}$ can operate with its own coverage radius r_i which varies in the range $[r_i^{min}; r_i^{max}]$.
- Let x_{ij} be a binary variable indicating if the edge (i, j) of G is considered in the final solution ($x_{ij} = 1$), or not ($x_{ij} = 0$).
- Let $\mathcal{N}(i)$ be the set of neighborhood nodes/antennas of i. A node j is a neighbor of i only if the condition (4.1) used with the maximum radii values, is verified.
- Let $\mathcal{I}(i, j)$ be the set of all edges (i, j) that don't intersect with any other edge (k, l), where (i, j) and (k, l) do not have common extremities.

The objective of the full network coverage problem is to detect rapidly holes in the network and reduce considerably inter-cell interference represented by the overlapping regions measured mathematically by formula (4.3). These objectives will be reached by optimizing the radii values of the antennas. This is equivalent to select in the final solution, only couple of antennas with a minimum Euclidean

distance guaranteeing the graph connectivity and the full network coverage which allows to intuitively reduce the inter-cell interference. This objective is given by:

$$\min \Gamma = \sum_{i \in \mathcal{A}} \sum_{j \in \mathcal{N}(i)} d_{ij} \times x_{ij} \tag{4.4}$$

The full network coverage problem has to comply with a number of constraints which will be summarized and mathematically expressed in the following.

Constraints (4.5) guarantee that each node i has at least two neighbors in the graph (we recall that the objective is to obtain a triangulation meeting connectivity and reduced interference).

$$\sum_{j \in \mathcal{N}(i)} x_{ij} \geq 2, \forall i \in \mathcal{A} \tag{4.5}$$

Constraints (4.6) impose that if an edge (i, j) do not have any intersection in the initial graph (see solid line edges in Figure 4.3), then $x_{ij} = 1$ in the final graph (the solution graph). These constraints are mathematically provided by:

$$x_{ij} = 1, \forall i \in \mathcal{A}, \forall j \in \mathcal{N}(i), \forall (i, j) \in \mathcal{I}(i, j) \tag{4.6}$$

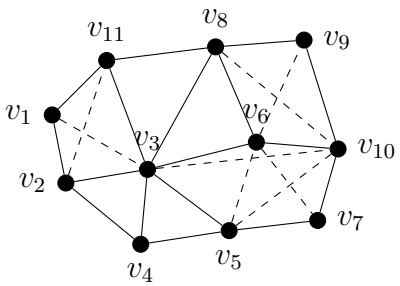


Figure 4.3: Each solid line edge (i, j) is necessary in the final graph/solution

In order to avoid any intersection in the final graph between any two edges (i, j) and (k, l) (see Figure 4.4 in which i, j, k, l can be represented by v_1, v_6, v_4, v_5 respectively), we propose the following nonlinear inequality:

$$x_{ij} \times x_{il} \leq x_{jl} + \sum_{k \in \mathcal{N}(i)} x_{ik}, \forall i \in \mathcal{A}, \forall j \in \mathcal{N}(i), \forall l \in \mathcal{N}(i) \cap \mathcal{N}(j) \tag{4.7}$$

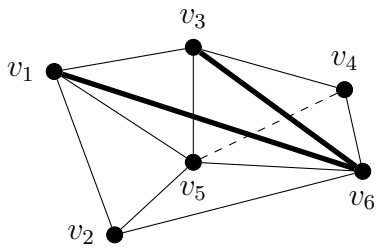


Figure 4.4: Example of edge intersection (interference)

Constraints (4.7) are nonlinear as we used the product of two decision variables. We replace them (constraints (4.7)) by new family of linear inequalities when introducing a new binary variable z_{ijl}, such that $z_{ijl} = x_{ij} \times x_{il}$, $\forall i \in \mathcal{A}, \forall j \in \mathcal{N}(i), \forall l \in \mathcal{N}(i) \cap \mathcal{N}(j)$. Thus, we obtain the constraints in (4.8), (4.9) and (4.10).

$$z_{ijl} \leq x_{ij} \tag{4.8}$$

$$z_{ijl} \leq x_{il} \tag{4.9}$$

$$z_{ijl} \geq x_{ij} + x_{il} - 1 \tag{4.10}$$

By summing (4.8) and (4.9), we obtain:

$$z_{ijl} \leq \frac{1}{2}\left(x_{ij} + x_{il}\right)$$

We finally have three new valid inequalities for the full network coverage problem, and they are provided by:

$$z_{ijl} \leq x_{jl} + \sum_{k \in \mathcal{N}(i)} x_{ik}, \forall i \in \mathcal{A}, \forall j \in \mathcal{N}(i), \forall l \in \mathcal{N}(i) \cap \mathcal{N}(j) \tag{4.11}$$

$$z_{ijl} \leq \frac{1}{2}(x_{ij} + x_{il}), \forall i \in \mathcal{A}, \forall j \in \mathcal{N}(i), \forall l \in \mathcal{N}(i) \cap \mathcal{N}(j) \tag{4.12}$$

$$z_{ijl} \geq x_{ij} + x_{il} - 1, \forall i \in \mathcal{A}, \forall j \in \mathcal{N}(i), \forall l \in \mathcal{N}(i) \cap \mathcal{N}(j) \tag{4.13}$$

Our mathematical model is hence characterized by the following Integer Linear Programming:

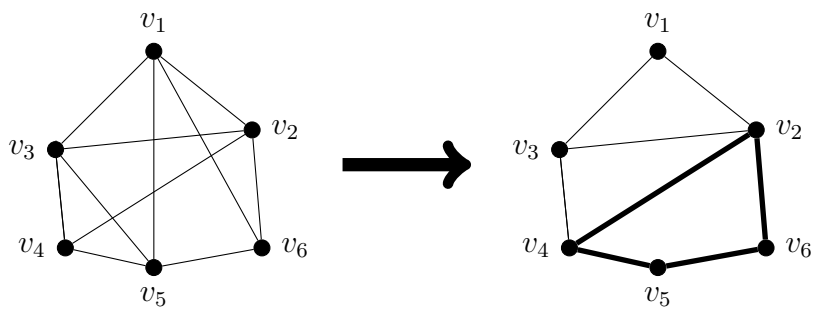
$$\begin{aligned}
&\min \Gamma = \textstyle\sum_{i \in \mathcal{A}} \sum_{j \in \mathcal{N}(i)} d_{ij} \times x_{ij} \\
&S.T.: \\
&\textstyle\sum_{j \in \mathcal{N}(i)} x_{ij} \geq 2, \forall i \in \mathcal{A} \\
&x_{ij} = 1, \forall i \in \mathcal{A}, \forall j \in \mathcal{N}(i), \forall (i,j) \in \mathcal{J}(i,j) \\
&z_{ijl} \leq x_{jl} + \textstyle\sum_{k \in \mathcal{N}(i)} x_{ik}, \forall i \in \mathcal{A}, \forall j \in \mathcal{N}(i), \forall l \in \mathcal{N}(i) \cap \mathcal{N}(j) \\
&z_{ijl} \leq \tfrac{1}{2}(x_{ij} + x_{il}), \forall i \in \mathcal{A}, \forall j \in \mathcal{N}(i), \forall l \in \mathcal{N}(i) \cap \mathcal{N}(j) \\
&z_{ijl} \geq x_{ij} + x_{il} - 1, \forall i \in \mathcal{A}, \forall j \in \mathcal{N}(i), \forall l \in \mathcal{N}(i) \cap \mathcal{N}(j) \\
&x_{ij}, z_{ijl} \in \{0,1\}, \forall i \in \mathcal{A}, \forall j \in \mathcal{N}(i), \forall l \in \mathcal{N}(i) \cap \mathcal{N}(j);
\end{aligned} \tag{4.14}$$

4.3.2 New valid inequalities

To address larger problem instances and to better describe the convex hull of the full network coverage problem, we need to investigate new valid inequalities and facets allowing to accelerate convergence time and to find optimal solutions jointly. Thus, we propose to investigate new families of inequalities that are valid for our problem.

4.3.2.1 Chordless cycle inequalities

Solving the mathematical formulation provided in (4.14) allows to obtain optimal solutions for the full network coverage problem. Nevertheless, and for some initial graph instances, the described convex hull in (4.14) is missing some solutions that do not contain holes. In Figure 4.5, we show a simple example of cellular network composed by 6 antennas with various coverage radii that can be represented by the left graph. The solution obtained (the right graph) by the ILP formulation in (4.14) has a coverage hole $(v_2, v_4, v_5, v_6, v_2)$.



(a) Initial Graph (b) Final Graph

Figure 4.5: Example of graph solution containing a hole $(v_2, v_4, v_5, v_6, v_2)$

Therefore, to integrate holes detection constraints in our mathematical formulation, we investigate a new facet or valid inequality based on holes detection that should be added to our optimization. These inequalities are based on detecting **Chordless Cycles**, that will be defined in the following.

Definition 4.3.1 *[105] Let G be an undirected graph and let $v_0, v_1, \ldots, v_{k-1}$ be a sequence of k distinct vertices such that there is an edge between v_i and $v_{(i+1) \bmod k}$ $(\forall i = 0, \ldots, k-1)$, and no other edge between any two of these vertices. Then, this sequence is a chordless cycle on k vertices. A hole may be a chordless cycle on four or more vertices.*

According to the previous definition, we would like to optimally solve the full network coverage problem when detecting all the existing holes in the initial graph. For this, we propose the following result.

Theorem 4.3.1 *For any initial connected graph G, and for each chordless cycle C in G, such that $|C| \geq 4$, the following inequality (4.15) is valid for the global*

coverage hole problem ($E(C)$ is the set of edges of the chordless cycle C):

$$x(E(C)) \leq 3 \tag{4.15}$$

Proof Let G be an undirected graph and $v_0, v_1, \ldots, v_{k-1}$ (with $k \geq 4$) the set of vertices making a chordless cycle (i.e. a hole) noted by C. Our objective is to detect chordless cycles (holes) and then eliminate them using our optimization.

Using the absurd reasoning, we suppose that $x(E(C)) = \sum_{i=0}^{k-2} x_{v_i,v_{i+1}} \geq 4$ which means that our optimization should keep at least 4 edges in C leading to obtain one of the two following cases:

1. A solution with a chordless cycle noted by $\{v_0, v_1, \ldots, v_{k-1}, v_0\}$ which represents a hole

2. A solution with at least two intersecting edges that represent an interference in the final solution

The case 1 is not feasible as our optimization is focusing on eliminating all the existing holes. The second case 2 cannot hold thanks to constraints (4.7) eliminating intersections and interference efficiently.

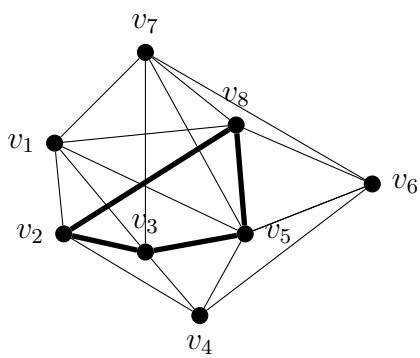


Figure 4.6: Example of a chordless cycle $(v_2, v_3, v_5, v_8, v_2)$ of size 4

To better understand the proof, we propose a simple example of cellular network represented by the graph of Figure 4.6 that contains a chordless cycle $C = \{v_2, v_3, v_5, v_8, v_2\}$ of size 4 . We can easily remark that (v_2, v_3), (v_3, v_5) and (v_5, v_8) do not intersect with any other edge in the graph. So, by applying constraints (4.6), we obtain a solution with $x_{v_2v_3} = 1$, $x_{v_3v_5} = 1$ and $x_{v_5v_8} = 1$. Hence, we discuss two cases on the status of the edge (v_2, v_8) of the chordless cycle C:

- $x_{v_2v_8} = 1$: In order to eliminate intersections in the final graph/solution, the edges (v_1, v_3), (v_1, v_5), (v_3, v_7) and (v_5, v_7) will be removed ($x_{v_1v_3} = 0$, $x_{v_1v_5} = 0$, $x_{v_3v_7} = 0$ and $x_{v_5v_7} = 0$) using constraints (4.7). This means that our final solution has a coverage hole, and this is not desirable.

- $x_{v_2v_8} = 0$: There is no coverage hole in the final graph/solution (full coverage) while totally eliminating network interference. In this case $x_{v_2v_3} + x_{v_3v_5} + x_{v_5v_8} + x_{v_8v_2} \leq 3$, leading to $x(E(C)) \leq 3$.

■

4.3.2.2 Separation of chordless cycles inequalities (4.15)

Thanks to inequalities (4.15), we guarantee the non existence of holes in our final and optimal solution. This is due to the generation and implication of (4.15) in the mathematical model (4.14). Nevertheless, as the number of chordless cycles can be exponential, then we cannot explore all of the existing chordless cycles as this can be time consuming for our optimization.

The separation of inequalities (4.15) consists in finding chordless cycles C^* violating constraints (4.15). This is a well known NP-Hard problem [45] and there is an exponential number of possibilities. Thus, we only explore finding few number of chordless cycles to eliminate holes in the final graph.

In order to find sufficient number of chordless cycles, we use an approximation algorithm that recursively executes Depth First Search (DFS) method. In fact, the main idea of this algorithm is to generate, for each vertex, a set of expanding paths using DFS strategy until a chordless cycle is found (i.e. the selected path should respect the conditions in Definition 4.3.1). This algorithm provides a set of violated chordless cycles constraints in acceptable times (in order of $\boldsymbol{O(n+m^2)}$, where n and m are respectively the number of vertices and edges in the graph G). We add these facets to our final optimization to eliminate possible holes in the final graph. For further details about the heuristic algorithm that we use to enumerate some chordless cycles, see [105] and [106].

4.3.2.3 Connectivity inequalities

In addition to the constraints eliminating chordless cycles (4.15), the new mathematical formulation (4.14) + (4.15) can lead to find optimal triangulations in a non connected graph (see Figure 4.7 representing a triangulation of two connected components).

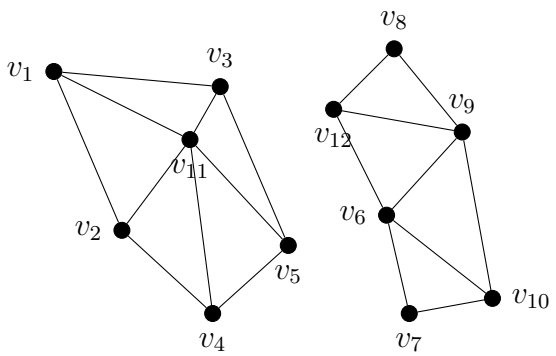


Figure 4.7: Example of two connected components (triangulations) creating holes in the final graph

Moreover and by definition, two connected components in the final graph are creating a hole. Thus, we investigate new valid inequalities (facets) to obtain a unique optimal triangulation without holes. This consists to guarantee the connectivity of

the final graph. We introduce the following constraints that will be integrated to our mathematical formulation.

Theorem 4.3.2 *For any initial graph $G = (\mathcal{A}, \mathcal{E})$, and for each subset $S \subseteq \mathcal{A}$, the following inequality (4.16) is valid to guarantee the network connectivity for the global coverage hole problem:*

$$x(\delta(S)) \geq 1 \tag{4.16}$$

where $\delta(S)$ represents the set of edges with exactly one extremity (or end-point) in S and the other one in the complement set of S (i.e. $\overline{S}$).

Proof Let a and b two nodes or antennas in the final graph that contains at least two connected triangulations components. For instance, we can imagine that $a = v_3$ and $b = v_8$ as illustrated in Figure 4.7. Then, it is clear that the maximum flow or the minimum cut between a and b in the graph of Figure 4.7 is zero, as they are separated into two connected components. The objective in our problem is to construct one triangulation with a minimum weight when guaranteeing the connectivity. Thus, we impose a and b to be in the same component. To do this, we simply have to impose that the maximum flow or the minimum cut between this couple of nodes should be greater than 1 (the value 1 is selected to guarantee that there exists at least **one** edge between a and b). Recall that the considered weights in this graph are the actual solution $x_e, e \in \mathcal{E}$ of the full network coverage problem. By applying connectivity constraints (4.16) we guarantee that all separated couples of nodes will be jointly on the same and unique connected component.

■

4.3.2.4 Separation of connectivity inequalities (4.16)

The separation problem of (4.16) consists in finding the optimal set of nodes (antennas) S^* that violates these constraints. Thus, we investigate all of the possible couples of nodes a and b that are not in the same connected component in the final graph. Next to that, we identify a minimum cut (set of edges) separating a and b, and then impose that the value of this minimum cut (using the weights x) will not exceed 1. Exploring all of the possible sets S violating (4.16) is NP-Hard as there is an exponential number of possibilities. Thus, we propose to explore only few number of sets that can be found polynomially when solving the minimum cut or maximum flow problem using a well known algorithm such Ford-Fulkerson [107]. In fact, few generations of (4.16) can be sufficient to guarantee the connectivity of the final graph.

4.3.3 Complete mathematical formulation

To summarize, and by considering all of the described constraints leading to find optimal solutions for the full network coverage problem, our mathematical formula-

tion is then provided by:

$$
\begin{aligned}
&\min \Gamma = \textstyle\sum_{i\in\mathcal{A}} \sum_{j\in\mathcal{N}(i)} d_{ij} \times x_{ij} \\
&S.T.: \\
&\textstyle\sum_{j\in\mathcal{N}(i)} x_{ij} \geq 2, \forall i \in \mathcal{A} \\
&x_{ij} = 1, \forall i \in \mathcal{A}, \forall j \in \mathcal{N}(i), \forall (i,j) \in \mathcal{I}(i,j) \\
&z_{ijl} \leq x_{jl} + \textstyle\sum_{k\in\mathcal{N}(i)} x_{ik}, \forall i \in \mathcal{A}, \forall j \in \mathcal{N}(i), \forall l \in \mathcal{N}(i) \cap \mathcal{N}(j) \\
&z_{ijl} \leq \tfrac{1}{2}(x_{ij} + x_{il}), \forall i \in \mathcal{A}, \forall j \in \mathcal{N}(i), \forall l \in \mathcal{N}(i) \cap \mathcal{N}(j) \\
&z_{ijl} \geq x_{ij} + x_{il} - 1, \forall i \in \mathcal{A}, \forall j \in \mathcal{N}(i), \forall l \in \mathcal{N}(i) \cap \mathcal{N}(j) \\
&x(E(C)) \leq 3, \forall C \subseteq \mathcal{A}, |C| \geq 4, C \;\; is \;\; a \;\; chordless \;\; cycle \\
&x(\delta(S)) \geq 1, \forall S \subseteq \mathcal{A} \\
&x_{ij}, z_{ijl} \in \{0,1\}, \forall i \in \mathcal{A}, \forall j \in \mathcal{N}(i), \forall l \in \mathcal{N}(i) \cap \mathcal{N}(j);
\end{aligned}
\tag{4.17}
$$

Finally, we provide, in Algorithm 5, a summary of our proposed algorithm based on branch-and-Cut formulation to cope with the full network coverage problem. The final Algorithm 5 describes all steps, including graph construction and transformation and the execution of the complete mathematical formulation provided by (4.17).

Algorithm 5 Full network coverage algorithm : Branch-and-Cut approach

Input: A real telecommunications network (cells with antennas) with given interference and coverage holes

Output: A full coverage network (no holes) with no interference

- Graph transformation of the real network G: each antenna is a node
- There is an edge between two nodes (antennas) i and j if $r_i + r_j \geq d_{ij}$
- An interference is represented by an intersection of two edges
- Run the Branch and Cut optimization model (4.17)
- The optimized/obtained network has no holes and no interference

4.4 Performance evaluation

4.4.1 Simulation parameters and settings

The performance evaluation of our algorithm, coded in Java, is conducted using an Intel Core CPU at 2.40 GHz with 8 GB RAM. Each initial network represented by a graph comprises a random number of antennas following a Poisson process with a parameter $\Lambda = \lambda \times space_dimensions$, where λ is varying in the range $[0.1; 1]$, and $space_dimensions$ are generated according to two essential spaces (5×5 and 10×10). Each antenna, represented by a vertex of this graph, has a radius value initialized to $r_{max} = 1km$. The simulation considers the generation of 100 feasible instances for each run. For sake of clarity, we summarize the simulation settings and parameters in Table 4.1.

Table 4.1: Network coverage optimization : simulation settings and parameters

Parameters	**Values**
Density of Antennas	$\lambda \in [0.1; 1]$
Space Dimensions	5×5; 10×10;...
Poisson Parameter	$\Lambda = \lambda \times space_dimensions$
Number of Antennas	Poisson Distribution: $\mathcal{P}(\Lambda)$
Antenna Coordinates	Uniform Distribution: $\mathcal{U}(0, space_dimensions)$
Min Coverage Radius	$r_{min} = 0.1km$
Max Coverage Radius	$r_{max} = 1km$

For our simulation and experiments, we use the optimization solver Cplex [94] to solve the exact mathematical model (4.17), and we also compare and benchmark our algorithm to other existing approaches.

4.4.2 Performance metrics

The algorithm performance assessment is based on the following metrics:

- **Convergence time**: is the time needed by the proposed exact algorithm to find an optimal solution.
- **Interference elimination rate**: is the rate of eliminated interference such that, and with no loss of generality, we consider an interference as an intersection of two edges in the final graph (i.e. the triangulation graph).

- **Coverage hole**: is the network coverage in terms of existing holes in the final solution. Hence, zero holes leads to a **full** coverage hole.

To assess performance of the proposed approach of the full network coverage problem using the described metrics, we considered a real trace and random instances described as follows:

1. **Random instances**: networks with an average number of antennas ranging in $[7; 100]$ interval, and an average number of edges in the $[18; 456]$ interval according to the formula (4.1).

2. **Real trace**: we used a real trace from a coverage cell 4G-LTE of the network operator Orange, in a small area in Paris [3]. This topology is in an area containing 26 4G-LTE antennas with their given geographical positions (coordinates), 94 edges and a maximum radius value of 0.4 km for each antenna.

4.4.3 Simulation results and performance analysis

4.4.3.1 Algorithm's performance comparison with the state-of-the-art

Our performance evaluation starts by assessing the execution time needed by the Branch-and-Cut algorithm to find the optimal solution to the full network coverage problem.

Table 4.2: Exact algorithm performance: convergence time to the optimum

Space	λ	**#Antennas**	**#Edges**	**#(4.15)***	**#(4.16)****	**Convergence Time (s)**
5×5	0.3	7.5	18.81	1	1	0.016
	0.5	12.5	34.63	1.13	1	0.0677
	0.8	20	70.21	5.79	3.45	9.43
	1.0	25	105.4	11.92	10.25	34.31
10×10	0.3	30	96.16	6.08	5.81	4.28
	0.5	50	186.21	17.79	12.64	64.47
	0.8	80	298.24	32.64	5.71	67.13
	1.0	100	456.35	63	14.59	139.4

* #(4.15) is the average number of chordless cycle constrains (4.15) added to the mathematical formulation.
** #(4.16) is the average number of connectivity constrains (4.16) added to the mathematical formulation.

In Table 4.2, the average convergence time to the optimum remains bellow 35 seconds and 140 seconds in the worst case for the scenario with average networks size of 100 antennas. The Branch-and-Cut algorithm scales reasonably well with problem size (number of antennas and edges) and thanks to the efficiency of the added cutting

planes provided by formulas (4.15) and (4.16) guaranteeing an optimal result with $\beta_0 = 1$ and $\beta_1 = 0$ (a covered network without holes and with a unique connected component). Moreover, this average execution time depends on the average number of added constraints ((4.15) and (4.16)) to our global optimization. As illustrated in Table 4.2, running and adding these constraints is requiring negligible times thanks to the heuristics approaches deployed to separate them in polynomial time.

In the following, we assess the convergence time and interference elimination rate of our algorithm and benchmark them with the solution provided by Rips complex which is a well known method to cope with the network coverage hole problem. Recall that Rips complex is a simplicial homology-based approach that consists in verifying the intersections between cells to detect holes and connectivity problems. For further details about this approach, we provided in Section 2.2.2 of Chapter 2 an overview of simplicial homology approaches, including Rips complex.

As our approach is based on an exact mathematical formulation leading to always attend the optimum, then Rips-based approach can be considered as an upper bound to our algorithm.

Table 4.3: Performance comparison : ILP vs Rips approach

Space	Density	#Antennas	#Edges	Convergence Time (s)		Interference elimination (%)	
				ILP	Rips*	ILP	Rips*
5×5	0.3	7.5	18.81	0.016	0.67	100	88.89
	0.5	12.5	34.63	0.0677	2.51	100	96.42
	0.8	20	70.21	9.43	12.36	100	97.91
	1.0	25	105.4	34.31	34.76	100	**98.01**
10×10	0.3	30	96.16	4.28	11.83	100	96.15
	0.5	50	186.21	64.47	**57.38**	100	95.61
	0.8	80	298.24	67.13	**63.91**	100	94.96
	1.0	100	456.35	139.4	**74.9**	100	97.97

* Rips-based approach is one of the most efficient algorithms that use Simplcial homology techniques to deal with the full network coverage problem (for more details, see Section 2.2.2 of Chapter 2).

For small graphs or networks (at most 25 antennas in Table 4.3), the Branch-and-Cut algorithm has negligible average convergence time compared to the necessary convergence time required by Rips method. The worst case for these graphs concerns the scenario with 25 antennas in which the ILP necessitates 34.31 seconds (to converge to the **optimal solution**) compared to 34.76 seconds for Rips method to converge to a **feasible solution**.

For larger graphs (between 50 and 100 antennas in Table 4.3) our algorithm is spending little more time compared to the convergence time of Rips method, as

we spent time to reach the optimum in the contrary of Rips method looking only for a feasible solution, and not necessary optimal ones. This is confirmed by the interference elimination rates provided in Table 4.3.

The interference elimination rate metric is reported in Table 4.3 and confirms that the Branch-and-Cut algorithm performs better than Rips-based approach even for large networks. In fact, our approach is eliminating **totally** the interference for all the considered scenarios compared to Rips-based approach that proposes final solutions with remaining interference and holes. Thus, our proposed approach guarantees the optimality of the found solution in terms of interference elimination and full network coverage jointly. Rips method is providing weak network coverage hole and **partial** interference elimination (98.01% as the best result when considering small graphs). Hence, by considering jointly the expected convergence time, the total interference elimination and the full network coverage solutions, the Branch-and-Cut approach can be used online by network providers offering connectivity and mobile services to end-users.

In other words, constraints (4.7) are dedicated to eliminating intersections and they are violated when two edges in the new graph, have a common point of intersection in the graph representation. Constraints (4.7) are stronger when combined with the other valid inequalities described in the mathematical model (4.17) which finds an optimal graph with only adjacent triangles (a full covered zone according to the Delaunay definition). Thus, the joint optimization leads to eliminate totally these intersections (interference) as the used Branch and Cut approach is guaranteeing the optimality (zero interference and no holes).

In the following, we evaluate the performance of our proposed Branch-and-Cut algorithm when considering a real cellular network and we analyze its scalability when addressing very large instances of the full network coverage problem.

4.4.3.2 Algorithm's performance evaluation using real traces

To better evaluate the performance of our exact algorithm based on Branch-and-Cut method, we consider a real cellular network as shown in Figure 4.8. This network is in a small area in Paris, containing 26 antennas, 94 edges and an interference rate equivalent to 80.85%. Note that in our work, and with no loss of generality, an interference is the intersection of two edges in the graphic representation of the network. In this experimentation, we would like to apply our Branch-and-Cut algorithm on the map of Figure 4.8 when assessing the three metrics cited above (i.e. coverage hole, interference elimination, and convergence time).

Figure 4.9 reveals for the topology in Figure 4.8 of reasonable size, the obtained covered network when applying our Branch-and-Cut algorithm which has the advantage of exploring the entire network space at once during optimization. The obtained triangulation in Figure 4.9 is optimal and with no holes leading to a network with a full network coverage. Our exact algorithm has totally eliminated the existing interference and reached the optimal solution in less than 1 sec (or exactly in 0.006 sec). This real network instance is in fact "easy" to solve using our Branch-and-Cut approach.

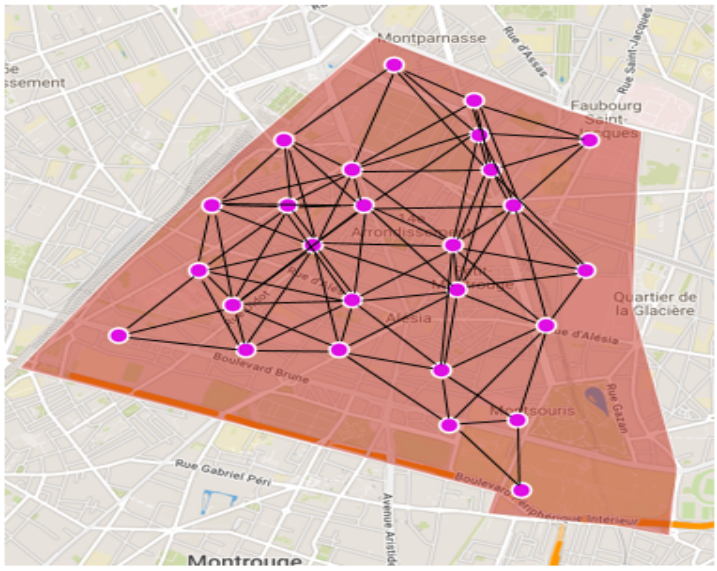


Figure 4.8: An Orange 4G-LTE cell map: **before Branch-and-Cut optimization**

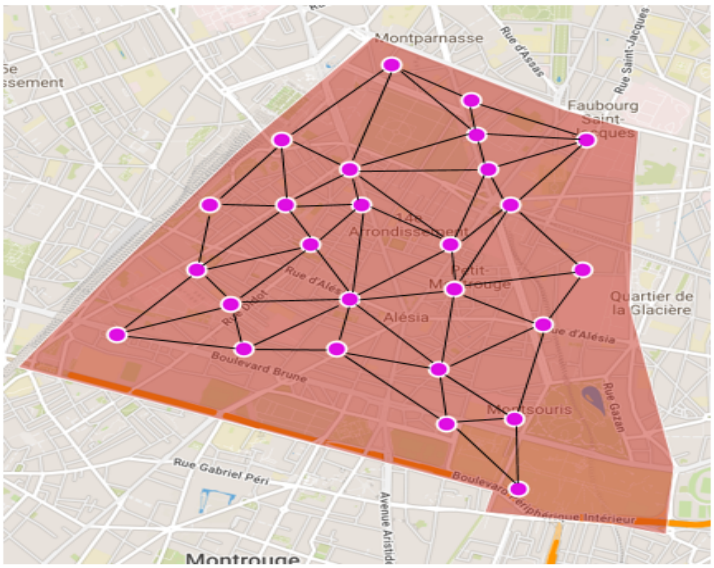


Figure 4.9: An Orange 4G-LTE cell map: **after Branch-and-Cut optimization**

4.4.3.3 Scalability analysis

To discuss the scalability analysis of our approach, we propose a network instance of 1000 antennas generated as described in Table 4.1. We apply our mathematical formulation provided by (4.17) and the obtained convergence time is close to **116.88 seconds** for an optimal solution with no holes (full coverage) and no interference. Note that the selected network/graph instance do not contain chordless cycles and the obtained result is a connected graph (i.e. without many connected components). This allows to avoid generating chordless cycles inequalities (4.15) and connectivity constraints (4.16) which can be time consuming when added to our optimization. Indeed, the generation of these cutting planes can be time consuming even if their used separation algorithms are converging in polynomial time, as it is depicted in the previous simulations in Table 4.2. This explains the necessary convergence time (for a network of 1000 antennas) which is less than the necessary time for a network with 100 antennas in which 63 chordless cycles constraints and 15 connectivity constraints are used to attend the optimum (see Table 4.2).

4.5 Conclusion

In this chapter, we proposed an exact mathematical formulation based on Branch-and-Cut methods to deal with the increase of inter-cell interference caused by high density of cells in C-RAN when maintaining the full network coverage. Then and in addition to classical mathematical modeling, we investigated new valid inequalities, i.e. chordless cycles and connectivity constraints, for our optimization model in order to scale with large number of antennas.

We evaluated the performance of our proposed approach using several simulation scenarios and a real network map. The simulation results reveal the efficiency of our approach that performs consistently well across all scenarios and performance metrics. This confirms the ability of our algorithms in providing good solutions that jointly optimize the full network coverage and minimize the inter-cell interference caused by network densification.

However, the dense deployment of antennas in C-RAN has another consequence which is the significant increase of the baseband processing amount needed to meet antenna demands. Since in C-RAN the baseband processing of antennas is carried out in centralized BBU pools, this imposes excessive bandwidth constraints and low latency requirements on the fronthaul network connecting RRHs to the centralized data centers. These requirements can be reduced by considering a more flexible split of baseband processing between RRHs and centralized BBU pool. The next chapter will focus on proposing new algorithms to optimally deploy BBU functions in C-RAN architecture when considering different split configurations and transport network characteristics.

Chapter 5

BBU function split placement in C-RAN

5.1 Introduction

With the growth in mobile data traffic demands, network operators will have to add significant amounts of spectrum as well as to increase the density of cells by deploying more antennas. This will not only increase the inter-cell interference levels (discussed in the previous chapter), but also will significantly increase the amount of baseband processing needed to meet the growing number of antenna demands and thus, increasing the amount of data traffic demands, with strict latency and bandwidth constraints, between antennas and centralized data centers. In fact, while we investigated in the previous chapter new solutions to reduce inter-cell interference when guaranteeing a full network coverage, this chapter discusses how to overcome the tradeoff between benefits of baseband processing centralization in BBU pools and strong latency requirements on fronthaul links.

In this chapter, we investigate new approaches to find best tradeoffs between centralization and transport requirements on fronthaul network. Indeed, to meet these requirements, more flexible distribution (or split) of baseband processing functions will be considered between RRHs and the centralized data centers. In this context, a range of BBU function splits is being introduced and studied, each of which presents different needs for capacity, latency and bandwidth on the fronthaul network (see chapter 2 for more details). In our work, we will consider 3GPP RAN split option [108] to model the aggregated antenna demands and we seek to efficiently determine the optimal placement of BBU functions in the centralized data centers based on the considered split configuration and transport network characteristics. Resource utilization gains can be achieved by determining the optimal placement of BBU functions when jointly meeting processing and latency requirements of the antennas demands.

To cope with this problem, we propose a mathematical model based on integer linear programming approach to optimally deploy the baseband processing function in the network when jointly minimizing the resource consumption and the expected latency. Then, for sake of scalability, we investigate new heuristic algorithms based

on the construction of a multi-stage graph to obtain good solutions for larger network size in acceptable times. Finally, we evaluate, using several simulation scenarios, the efficiency of the proposed algorithms in finding optimal solutions with reduced complexity.

This chapter will be organized as follows. In Section 5.2, we introduce the system model used to address the BBU function split placement problem and we provide a brief discussion on the problem complexity. Section 5.3 describes a complete mathematical formulation of the addressed problem, based on an ILP approach for small problem size while in Section 5.4, we introduce four heuristic algorithms based on graph theory to accelerate the convergence time when guaranteeing good solutions. Section 5.5 reports the numerical results of performance evaluation highlighting the efficiency and scalability of the proposed algorithms. Finally, Section 5.6 concludes this chapter.

5.2 Problem statement

The BBU function split placement problem consists in determining the optimal locations of baseband functions in C-RAN network when considering the 3GPP RAN split option (a detailed overview of this functional split option with a description of considered BBU functions can be found in Section 2.3.3 of Chapter 2). The optimal placement of BBU functions can be achieved by finding the best tradeoff between BBU computation centralization and fronthaul network requirements while the network resource consumption is minimized.

In the following, we introduce the modeling of antennas demands by directed chains when considering 3GPP RAN split option and we describe the network topology which is considered to deploy BBU functions. Then, we present the system model considered to optimally solve the BBU function split placement problem and study the complexity of the addressed problem.

5.2.1 BBU function split modeling

We consider 3GPP RAN split to model our BBU function split placement problem. This split configuration is outlined as the best option in the 3GPP standards [108]. As depicted in Section 2.3.3 of Chapter 2, this split option consists in separating the baseband processing of antennas demands into three connected components : the first component is the PHY layer, the second contains the MAC and RLC layers and the third component for PDCP layer. In fact, PHY, RLC and MAC functions require lower fronthaul latency, and their processing requirement is dependent on the traffic. Thus, co-locating these functions on the infrastructure, shared across multiple cell sites, is expected to yield high pooling gains and also enable advanced techniques to manage inter-cell interference. On the contrary, the PDCP layer is less capacity-intensive and not subject to real-time constraints. Therefore, the PDCP layer can be placed flexibly to achieve increased multiplexing gains. Figure 5.1 shows the modeling of antennas demands according to this split option.

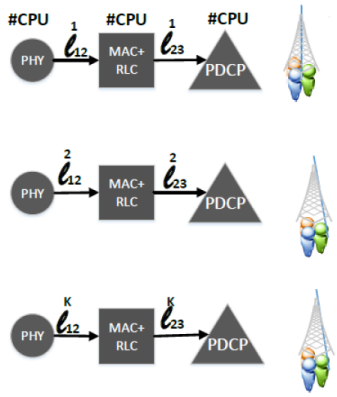


Figure 5.1: BBU function split modeling for each antenna demand

3GPP RAN split option is used to model the aggregated antenna demands which are represented, in Figure 5.1, by a set of directed chains. Each function chain is composed by three nodes representing four connected layers : PHY, MAC+RLC and PDCP. For sake of simplicity, each node has computing resource requirements, expressed in terms of CPU cores. These nodes are connected by two weighted arcs which indicate the sequencing between BBU functions with expected communication latency.

5.2.2 Network topology description

Figure 5.2 shows the physical network architecture that we will consider in the BBU function split placement problem. In fact, we model our physical network as an hierarchical architecture, which is a promising approach to achieve flexible deployment strategies of BBU functions across shared data centers.

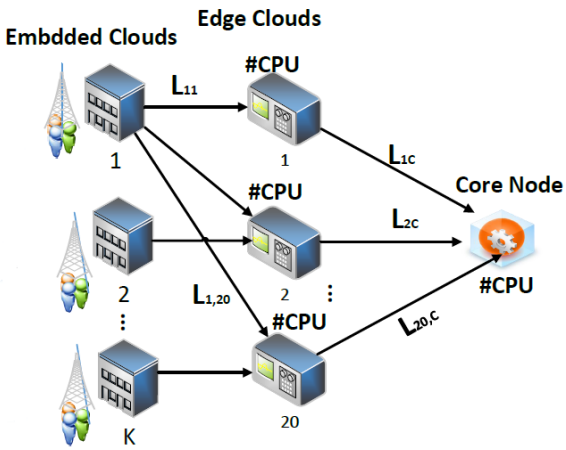


Figure 5.2: Physical network architecture

Our network architecture is composed by a set of embedded cloud data centers, a limited number of edge cloud data centers, and a centralized core cloud data center,

each of which has its own computing processing resources represented by number of available CPU cores. The embedded clouds are located close to the antennas and can be supported by accelerators (DSP and FPGA). Hence, the embedded cloud will be probably employed for executing PHY functions. We note that each embedded cloud is assigned to exactly one antenna and thus the number of antennas is equal to the number of embedded clouds. The edge cloud may be located further away from the antennas, and it is typically used for aggregating and processing traffic of multiple cell sites. Baseband functions above the cell-level PHY layer as well as functions for inter-site basedband coordination can be located at the edge cloud. The centralized core cloud is used for non real-time functions such as PDCP. For the best of our knowledge, this network modeling is very similar to Orange network topology, i.e. Next Generation Point of Presence (NG-PoP)[109].

5.2.3 System model

The aim of the BBU function split placement problem is to optimally deploy the requested chains (Figure 5.1) on the network architecture (Figure 5.2) when jointly meeting the CPU and latency requirements. Figure 5.3 illustrates our considered system model for the BBU function split placement problem.

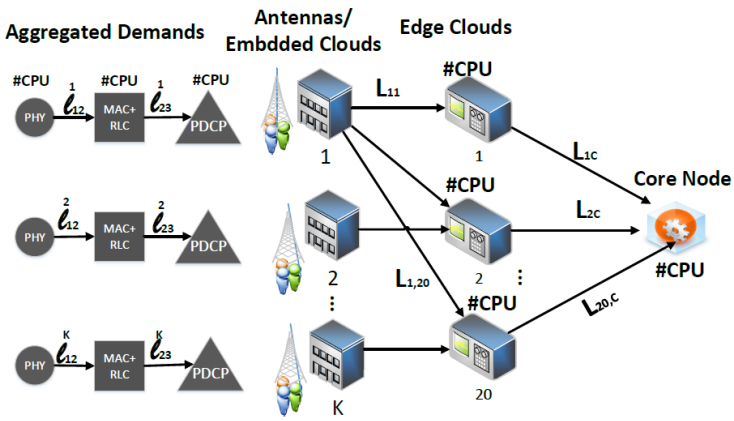


Figure 5.3: System model for BBU function split placement

Our system model contains a set of antennas/RRHs, denoted by $\mathcal{K}$, each of which has an aggregated demand represented by a directed chain (the left part of Figure 5.3). In fact, each chain has exactly 3 virtual nodes representing PHY, MAC+RLC and PDCP layers, and 2 virtual arcs to represent the BBU functions chaining. Each virtual node i of antenna demand k has variable processing requirements expressed in terms of the number of CPU cores denoted by c_i^k. Each virtual arc $(i, i+1)$ connecting two consecutive virtual nodes i and $i+1$ has a latency requirements, denoted by $l_{i,i+1}^k$. On the right part of Figure 5.3, we model the physical network as an undirected graph $G = (\mathcal{J}, \mathcal{E})$ where $\mathcal{V}$ and $\mathcal{E}$ are the sets of available physical nodes and arcs, respectively. Each physical node j, in which j can be an edge node, embedded node, or core node in the corresponding cloud, has a limited CPU capacity denoted by C_j. There is an arc between two physical nodes j and

j' of two different levels with a fronthaul latency denoted by $L_{j,j'}$. Note that if the physical node j' is not a neighbor of j, then the latency $L_{j,j'}$ is the sum of latency values on the shortest path between nodes j and j'. We denote by $\mathcal{P}(j)$, the set of all physical nodes j' such that there exists a shortest path between nodes j and j'.

For sake of clarity, we summarize in Table 5.1 all parameters and variables that are used to define our system model. These notations will also be used later in Section 5.3.

Table 5.1: BBU function split placement problem : variables and parameters

$G = (\mathcal{J}, \mathcal{E})$	:	weighted directed graph (the right part of Figure 5.3, for instance)
$\mathcal{J}$	:	set of physical nodes j including embedded clouds, edge clouds and one core cloud
$\mathcal{E}$	:	a set of communication links between different physical nodes j
C_j	:	Number of CPU cores **available** in the physical node j (embedded, edge and core clouds)
$L_{(j,j')}$	:	Latency on the link joining two physical nodes j and j'
$j' \in \mathcal{P}(j)$	:	set of all physical nodes that joins the physical node j to j'
$\mathcal{K}$	:	set of directed chains k that represents the aggregated demands (the left part of Figure 5.3)
$i \in \{1, 2, 3\}$	:	set of virtual nodes i of each chain k where 1 represents the PHY layer, 2 represents the second node which contains the MAC and RLC layers and 3 represents the PDCP layer
c_i^k	:	Number of CPU cores **requested** for processing the virtual node i (PHY, MAC + RLC and PDCP layers) of chain k
$l^k_{(i,i+1)}$	:	Expected latency between two consecutive virtual nodes i and $i+1$ of the same chain k

5.2.4 Problem complexity

Before investigating new algorithms, we discuss the complexity of the BBU function split placement problem. We note that the aim of this problem is to determine the optimal mapping of the requested chains on the multi-stage network when jointly meeting strong latency and CPU requirements of the aggregated demands. We introduce the following theorem for the complexity of the addressed problem:

Theorem 5.2.1 *The BBU function split placement problem is NP-Hard.*

Proof The aim of BBU function split placement problem is to optimally deploy the baseband processing functions of antenna demands on the network topology (see Figure 5.3) when jointly meeting latency and CPU requirements. By considering the system model described in Section 5.2.3, our problem consists in finding the optimal mapping of a set of directed chains (BBU functions) on multi-stage graph (physical network topology). This problem formulation is very close to the well known Virtual Network Embedding (VNE) problem (described in Figure 5.4).

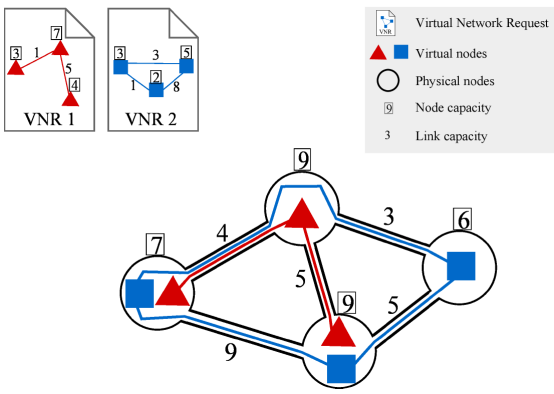


Figure 5.4: Example of Virtual Network Embedding problem. Source: [4]

In fact, the VNE problem consists in finding an optimal mapping of Virtual Network Requests (VNR) on a physical network when both computational and network requirements are met, which is equivalent to find a mapping of two undirected graph. This problem has been very well studied in the literature and its complexity has been investigated. In fact, authors in [110] proposed a general polynomial-time reduction which implies the NP-hardness of the VNE problem even when restricting the problem to very specific subclasses of graphs (for more details on VNE problem, see also [111] and [112]). Since our problem has additional constraints compared to VNE problem, such as the chaining of BBU functions, we deduce that the BBU function split placement problem is also NP-Hard.

■

5.3 Exact mathematical approach

In this section, we investigate a new mathematical formulation based on ILP approach to address the BBU function split placement. This problem consists in determining the optimal mapping of the requested chains on the physical network when optimizing the resource consumption in terms of CPU cores and the expected latency.

Decision variables

We start our problem's modeling by introducing three decision variables as follows:

- $x^k_{i,j}$ is a binary variable, the value of which is 1 if the virtual node, i.e. BBU function, i of the antenna k is placed on the physical node j, and 0 otherwise.
- $y^k_{(i,i+1);(j,j')}$ is a binary variable, the value of which is 1 if a virtual edge $(i, i+1)$ for a link between two BBU functions of the antenna demand k is placed on a physical path joining two physical nodes j and j', and 0 otherwise.
- z_j is a binary variable, the value of which is 1 if a physical node j is used, and 0 otherwise.

We recall that all parameters, which will be used in the ILP formulation, are summarized in Table 5.1 in the previous section.

Objective

The objective of the BBU function split placement is to map jointly all the connected chains to the physical network while minimizing the total CPU core consumption and the end-to-end latency. It is given by:

$$\min \quad \mathcal{F} = -\sum_{j\in\mathcal{J}}\left(C_j z_j - \sum_{k\in\mathcal{K}}\sum_{i\in\{1,2,3\}} c^k_i x^k_{i,j}\right) + \sum_{k\in\mathcal{K}}\sum_{j\in\mathcal{J}}\sum_{j'\in\mathcal{P}(j)}\sum_{i\in\{1,2\}} L_{(j,j')} y^k_{(i,i+1);(j,j')} \tag{5.1}$$

Formula (5.1) is the mapping from the left part of Figure 5.3 to the right part of Figure 5.3. The first term of (5.1) denotes the residual computing (CPU) resources in different data centers, i.e. embedded, edge and core cloud data centers while the second term represents the total costs in terms of latency provided for processing the BBU functions chains (the aggregated antenna demands). The optimum solution of the BBU function split placement problem will jointly reduce the expected latency on the fronthaul network and maximize the residual computing resources, in terms of CPU cores, in each node of physical network. This solution should respect a set of constraints of the addressed problem which will be detailed in the following.

Constraints

The optimization problem for the BBU function split placement has a certain set of constraints as follows:

- Constraints (5.2) avoid the placement of the physical layer of the chain k on the other embedded nodes that are not assigned to the antenna k.

$$\sum_{j\in\mathcal{J}_1} \mathbb{1}_{(k,j)} x^k_{1,j} = 0, \forall k \in \mathcal{K} \tag{5.2}$$

  where $\mathbb{1}_{(k,j)}$ is equal to 1 if the antenna k is not assigned to the embedded node j, and 0 otherwise and $\mathcal{J}_1$ represents the set of embedded clouds.

- Constraints (5.3) guarantee that each virtual node, i.e., a BBU function, is deployed on exactly one physical node.

$$\sum_{j \in \mathcal{J}} x_{i,j}^k = 1, \forall k \in \mathcal{K}, \forall i \in \{1,2,3\} \tag{5.3}$$

- Constraints (5.4) ensure that the placement of BBU functions cannot consume more resources than that available on the selected physical node.

$$\sum_{k \in \mathcal{K}} \sum_{i \in \{1,2,3\}} x_{i,j}^k \times c_i^k \leq C_j, \forall j \in \mathcal{J} \tag{5.4}$$

- Constraints (5.5) are used to guarantee the chaining of the BBU functions. For example, if the PHY layer is deployed on the physical node i, then the virtual node that contains the MAC and RLC layers should be deployed on j such that there is a **physical path** $P(j)$ starting from node i to node j.

$$x_{i,j}^k \leq \sum_{j' \in \mathcal{P}(j)} x_{i+1,j'}^k, \forall k \in \mathcal{K}, \forall i \in \{1,2\}, \forall j \in \mathcal{J} \tag{5.5}$$

- Constraints (5.6) and (5.7) are used to ensure that if a virtual node i is deployed on a physical node j, i.e. $x_{i,j}^k = 1$, and the virtual node $i+1$ is hosted by a physical node j', i.e. $x_{i+1,j'}^k = 1$, then the virtual arc $(i, i+1)$ should be deployed on the **physical path** starting from node j to node j', i.e. $y_{(i,i+1);(j,j')}^k = 1$.

$$\sum_{j \in \mathcal{V}} y_{(i,i+1);(j,j')}^k = x_{i+1,j'}^k, \forall k \in \mathcal{K}, \forall i \in \{1,2\}, \forall j' \in \mathcal{P}(j) \tag{5.6}$$

$$\sum_{j' \in \mathcal{P}(j)} y_{(i,i+1);(j,j')}^k = x_{i,j}^k, \forall k \in \mathcal{K}, \forall i \in \{1,2\}, \forall j \in \mathcal{J} \tag{5.7}$$

- Constraints (5.8) guarantee that each virtual arc $(i, i+1)$ is deployed on exactly one **physical path**.

$$\sum_{j \in \mathcal{J}} \sum_{j' \in \mathcal{P}(j)} y_{(i,i+1);(j,j')}^k = 1, \forall k \in \mathcal{K}, \forall i \in \{1,2\} \tag{5.8}$$

- Constraints (5.9) impose that the fronthaul latency on the selected path in the physical network must not exceed the latency requirements of the BBU function chains.

$$L_{(j,j')} \times y_{(i,i+1);(j,j')}^k \leq l_{(i,i+1)}^k, \forall k \in \mathcal{K}, \forall i \in \{1,2\}, \forall j \in \mathcal{J}, \forall j' \in \mathcal{P}(j) \tag{5.9}$$

- Constraints (5.10) indicate that if there exists at least one BBU function deployed on a physical node j, then the former should be used to host other virtual nodes if necessary.

$$x_{i,j}^k \leq z_j, \forall j \in \mathcal{J}, \forall k \in \mathcal{K}, \forall i \in \{1,2,3\} \tag{5.10}$$

Complete mathematical formulation

We summarize our mathematical model in the following ILP formulation (5.11). This ILP model uses Branch-and-Bound method (see Section 2.2.1.1 of Chapter 2) to provide for the BBU function split placement problem, the optimum solution from all possible ones.

$$
\begin{aligned}
\min \quad & \mathcal{F} = -\sum_{j\in\mathcal{J}} \left(C_j z_j - \sum_{k\in\mathcal{K}} \sum_{i\in\{1,2,3\}} c_i^k x_{i,j}^k \right) + \\
& \qquad \sum_{k\in\mathcal{K}} \sum_{j\in\mathcal{J}} \sum_{j'\in\mathcal{P}(j)} \sum_{i\in\{1,2\}} L_{(j,j')} y_{(i,i+1);(j,j')}^k \\
& S.T. : \\
& \sum_{j\in\mathcal{J}_1} \mathbb{1}_{(k,j)} x_{1,j}^k = 0, \forall k \in \mathcal{K} \\
& \sum_{j\in\mathcal{J}} x_{i,j}^k = 1, \forall k \in \mathcal{K}, \forall i \in \{1,2,3\} \\
& \sum_{k\in\mathcal{K}} \sum_{i\in\{1,2,3\}} x_{i,j}^k \times c_i^k \leq C_j, \forall j \in \mathcal{J} \\
& x_{i,j}^k \leq \sum_{j'\in\mathcal{P}(j)} x_{i+1,j'}^k, \forall k \in \mathcal{K}, \forall i \in \{1,2\}, \forall j \in \mathcal{J} \\
& \sum_{j\in\mathcal{J}} y_{(i,i+1);(j,j')}^k = x_{i+1,j'}^k, \forall k \in \mathcal{K}, \forall i \in \{1,2\}, \forall j' \in \mathcal{P}(j) \\
& \sum_{j'\in\mathcal{P}(j)} y_{(i,i+1);(j,j')}^k = x_{i,j}^k, \forall k \in \mathcal{K}, \forall i \in \{1,2\}, \forall j \in \mathcal{J} \\
& \sum_{j\in\mathcal{J}} \sum_{j'\in\mathcal{P}(j)} y_{(i,i+1);(j,j')}^k = 1, \forall k \in \mathcal{K}, \forall i \in \{1,2\} \\
& L_{(j,j')} \times y_{(i,i+1);(j,j')}^k \leq l_{(i,i+1)}^k, \forall k \in \mathcal{K}, \forall i \in \{1,2\}, \forall j \in \mathcal{J}, \forall j' \in \mathcal{P}(j) \\
& x_{i,j}^k \leq z_j, \forall j \in \mathcal{J}, \forall k \in \mathcal{K}, \forall i \in \mathcal{J}_v \\
& x_{i,j}^k \in \{0,1\}, \forall i \in \{1,2,3\}, \forall j \in \mathcal{J}; \\
& y_{(i,i+1);(j,j')} \in \{0,1\}, \forall i \in \{1,2\}, \forall j \in \mathcal{J}; \\
& z_j \in \{0,1\}, \forall i \in \{1,2\}, \forall j \in \mathcal{J};
\end{aligned}
\tag{5.11}
$$

The obtained solution by the ILP approach will be used as **reference and optimal solution** to evaluate the performance of the approximation algorithms that we will introduce, in the next section, to address large problem instances in acceptable times. This is not feasible with the exact approach based on ILP model (5.11) due to the NP-Hardness of our addressed problem.

5.4 Approximation approaches : multi-stage graph algorithms

To deal with large problem sizes, we propose efficient heuristic algorithms that converge to optimal or near-optimal solutions. These heuristics are based on the construction of an extended multi-stage graph $\mathcal{G}_m = (\mathcal{N}_m, \mathcal{E}_m, 3)$ by focusing on the number of available physical nodes, and the three BBU function layers. $\mathcal{E}_m$ is the set of arcs of $\mathcal{G}_m$ that will be clearly identified in the following.

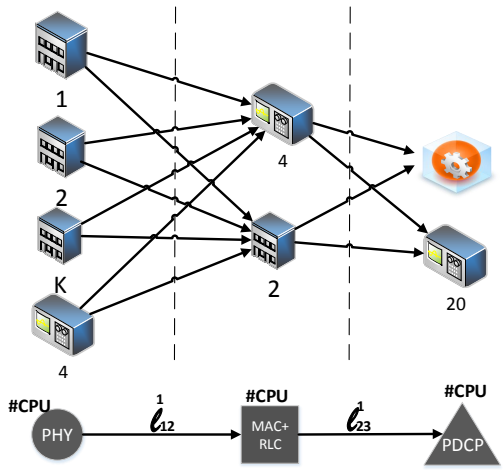


Figure 5.5: A multi-stage graph example

To populate the multi-stage graph $\mathcal{G}_m$, we suppose that each physical node, in the edge cloud and core cloud, is able to host different BBU functions (PHY, MAC+RLC, and PDCP) of different antennas and meeting the node's limited capacity in terms of CPU cores. Thus, from the multi-stage graph construction (Figure 5.5), the set of arcs between nodes j and j' in two different levels are weighted by the value of a shortest path $P_{j,j'}$ in terms of latency values on the links between physical nodes (using Dijkstra algorithm which is detailed in Section 2.2.1.4 of Chapter 2). We consider three levels in our multi-stage graph and each level corresponds to a BBU function.

Figure 5.5 represents the extended multi-stage graph that we obtain for an example of 4 available physical nodes in the first level, and two available nodes in the second and third levels. As shown in the multi-stag graph, we can find the same physical node at different levels depending on their available CPU capacity. We recall that each physical node can host more than one BBU function of the same BBU function chain.

Based on this graph, we propose **four** strategies of multi-stage approach to deploy BBU function chains on the available data centers.

- **MIN-MIN**: considers the BBU function chain that has the minimum total amount of CPU resources, i.e. the sum of CPU cores in each BBU function,

and deploys each BBU function, i.e. virtual node, on the physical node, i.e. data center, which has the **minimum** amount of available computing resources (CPU cores). This will be repeated until all BBU function chains are deployed on the physical network, otherwise, the problem has no complete solution using this strategy.

- **MIN-MAX**: selects the BBU function chain which has the **minimum** total amount of requested CPU resources and starts the placement of each virtual node by contacting the data center with the **maximum** amount of available CPU resources. This will be repeated until all BBU function chains are deployed on the physical network, otherwise, the problem has no complete solution using this strategy.

- **MAX-MIN**: considers the BBU function chain which has the **maximum** amount of requested CPU resources, and then places each BBU function on the physical node which has the **minimum** amount of available CPU resources. This will be repeated until all BBU function chains are deployed on the physical network, otherwise, the problem has no complete solution using this strategy.

- **MAX-MAX**: selects the BBU function chain which has the **maximum** total amount of requested CPU resources, and deploys each BBU functions on the physical which has the **maximum** amount of available CPU resources. This will be repeated until all BBU function chains are deployed on the physical network, otherwise, the problem has no complete solution using this strategy.

According to the previous strategies, we summarize the multi-stage approach in Algorithm 6.

Algorithm 6 Multi-stage graph algorithm

Input: Aggregated BBU functions chains, Physical network.

Output: A joint mapping of all the requested chains on the physical network.

Step 1: Select a strategy (MIN-MIN, MIN-MAX, MAX-MIN or MAX-MAX) to find the optimal or near-optimal mapping of BBU function chains;

Step 2: Sort all the requested chains according to the total amount of requested CPU for each chain c;

Step 3: Create the multi-stage graph $\mathcal{G}_m$ according to the description given above;

Step 4: If the selected chain c is deployed : repeat until all chains are deployed, otherwise : the problem has no **complete*** solution.

*There is a solution when all the BBU functions chains are deployed successfully.

Multi-stage based algorithm's complexity

It is important to evaluate the complexity of our proposed multi-stage algorithm. We note that the addressed problem is NP-Hard, and we need rapid and cost-efficient approaches to cope with this complexity. As described in Algorithm 6, we detail below the complexity of our algorithm :

- Step 1 is just used to select which strategy we will follow among MIN-MIN, MIN-MAX, MAX-MIN and MAX-MAX strategies.
- Step 2 consists in sorting the aggregated demands according to the total amount of requested CPU cores. For that, we used the well known "Quicksort method" with a complexity of $n \ln(n)$ (in our case, n represents the number of antennas or aggregated demands).
- In Step 3, we construct a multi-stage graph for each demand, thus its complexity in the worst case is equal to the number of physical nodes which is negligible in our case.
- Finally, we need to iterate n times to deploy all aggregated demands on physical network.

Therefore, our proposed algorithm has a global complexity of $O\left(n \ln(n) + n\right)$ in the worst case. This complexity is negligible to address large network sizes of BBU function split placement problem.

5.5 Performance evaluation

In this section, we assess the performance of the proposed algorithms using two simulation scenarios. The four heuristic algorithms are benchmarked with the ILP solution for small and medium problem sizes. We also discuss the scalability of our heuristic algorithms when considering large network sizes.

5.5.1 Simulation parameters and settings

For our simulations, we consider a physical network topology similar to that in [77]. Specifically, we consider a random number of antennas ranging in $[20, 500]$, a number of edge clouds in the interval $[10, 20]$, and 1 core node according to **Orange network topology** (see [109] for more details). We then use the two following scenarios:

- Scenario 1: **Random Graphs**: For each BBU function, we randomly generate a CPU requirement from $[1, 9]$ CPU cores, and a random amount of available CPU cores ranging in $[10, 50]$ CPU cores for the embedded cloud, $[30, 80]$ CPU cores for the edge cloud, and $\{50, 100\}$ CPU cores for the core cloud. Moreover, the latency requirements on each arc of each BBU function chain as well as the fronthaul latency values are randomly generated according to [108] and [80].

- Scenario 2: **Euclidean Graphs**: The same parameters as described above are adopted except for the latency on the physical arcs which is generated based on the Euclidean distance[1].

For our simulation and experiments, the performance evaluation of the proposed algorithms is conducted using an Intel Core CPU at 2.40 GHz with 8 GB RAM. We used the IBM optimization solver Cplex [94] to solve the exact mathematical model in (5.11) and we implemented our algorithms in Java.

5.5.2 Performance metrics

In order to evaluate the performance of our heuristics compared to the ILP formulation algorithm used as "the reference and optimal solution", we define two performance metrics as follows :

- **Convergence time**: is the time needed by the algorithms to converge to their best solutions.

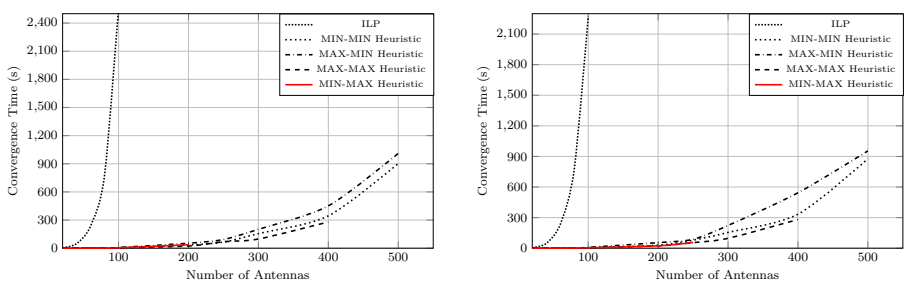
- **Gap**: is used to benchmark the proposed heuristics compared with the exact ILP formulation algorithm used as "the reference and optimal solution". This metric can be expressed as follows:

$$Gap(\%) = \frac{Cost_{heur} - Cost_{optimum}}{Cost_{optimum}} \tag{5.12}$$

  where $Cost_{heur}$ and $Cost_{optimum}$ are the objective functions (according to (5.1)) of the proposed heuristic approaches and the ILP solution, respectively.

5.5.3 Simulation results and performance analysis

Figures 5.6 depicts the convergence time of the ILP model and the four heuristic algorithms when considering Random and Euclidean graphs, respectively.



(a) **Euclidean graphs** (b) **Random graphs**

Figure 5.6: Algorithms' convergence time using 20 edge cloud data centers

[1]the distance between two points i and j with the coordinates (x_i, y_i) and (x_j, y_j) is provided by $\sqrt{(x_j - x_i)^2 + (y_j - y_i)^2}$) where the coordinates x and y are generated according to the positions of the nodes in Figure 5.3

The ILP solution is obtained based on the branch and bound algorithm that explores the convex hull of the BBU function split placement problem and then enumerates all the feasible solutions. This causes an exponential increase in terms of convergence time, which is undesirable.

Table 5.2 evaluates the quality of solutions obtained by heuristic algorithms by calculating the cost gap (according to Formula 5.12) for Euclidean and Random graphs. We note that the heuristic algorithm provides an **optimum solution** when the value of the **gap is equal to 0** in Table 5.2.

Table 5.2: Algorithms' performance comparison : ILP vs heuristic variants

#Antennas	#Edge Clouds	Heuristic Variant	Gap (Euclidean)* %	Gap (Random)** %
60	**10**	MIN-MIN	0	3.07
		MIN-MAX	0	0
		MAX-MIN	**2.96**	2.62
		MAX-MAX	0	0
	15	MIN-MIN	0	2.79
		MIN-MAX	0	0
		MAX-MIN	2.24	2.22
		MAX-MAX	0	0
	20	MIN-MIN	1.88	2.11
		MIN-MAX	0	0
		MAX-MIN	2.05	1.94
		MAX-MAX	0	0
80	**10**	MIN-MIN	2.49	**3.08**
		MIN-MAX	0	0
		MAX-MIN	2.64	2.56
		MAX-MAX	0	0
	15	MIN-MIN	2.92	2.76
		MIN-MAX	0	0
		MAX-MIN	2.3	2.32
		MAX-MAX	0	0
	20	MIN-MIN	2.55	2.28
		MIN-MAX	0	0
		MAX-MIN	1.87	2.0
		MAX-MAX	0	0

* The average gap (as described in 5.12) using scenario 2 : **Euclidean** graph.
** The average gap (as described in 5.12) using scenario 1 : **Random** graph.

As shown in Table 5.2, the *MAX-MAX* algorithm can provide optimal solutions (when the Gap = 0) of the problem in few seconds. This algorithm focuses on selecting chains according to the maximum amount of required CPU, and then choosing the available physical nodes with the maximum amount of CPU cores. Nevertheless, this algorithm is not able to explore much more alternatives or solutions in the space of feasible solutions when applied to large number of antennas. In fact, the *MAX-MAX* algorithm exploration is limited and can only address the cases with 400 antennas with 20 edge nodes as illustrated by Figure 5.6.

The *MIN-MAX* algorithm provides also optimal solutions (see Table 5.2 when the Gap = 0) for the BBU function split placement even in the case of a large

network size when we consider sufficient available resources. Unfortunately, this algorithm is not able to deeply explore the convex hull of the problem especially for the case of small number of edge nodes (see Figure 5.6).

The *MAX-MIN* and *MIN-MIN* algorithms provide near-optimal solutions. In fact, Table 5.2 shows that a maximum gap of 2.96% for Euclidean graphs, and 3.08% for Random graphs are achieved when considering algorithms *MAX-MIN* and *MIN-MIN* respectively. These algorithms are able to explore deeply the space of feasible solutions. For these reasons, *MAX-MIN* and *MIN-MIN* can address larger instances of BBU function split placement problem, compared to *MAX-MAX* and *MIN-MAX* approaches. In fact, Figure 5.6 depicts that *MAX-MIN* and *MIN-MIN* can provide solutions for our BBU function split placement problem when the number of antennas demands reaches 500 on a physical network containing 20 edge nodes while the *MAX-MAX* and *MIN-MAX* algorithms is able to solve problem instances of up to 400 antennas when considering 20 edge nodes.

In the following, we define residual resources (CPU) as the available and unused amount of servers' CPU using Euclidean graphs to assess the resource allocation of our heuristic algorithms. These algorithms are benchmarked by the ILP solution. We observe from Figure 5.7 that the solutions in terms of CPU residual resources obtained from the heuristic algorithms are close to those of the ILP solution. The physical nodes allocated by the heuristic algorithms are similar to those of the ILP solution. This is represented by negligible difference/gap between the curves illustrated in Figure 5.7.

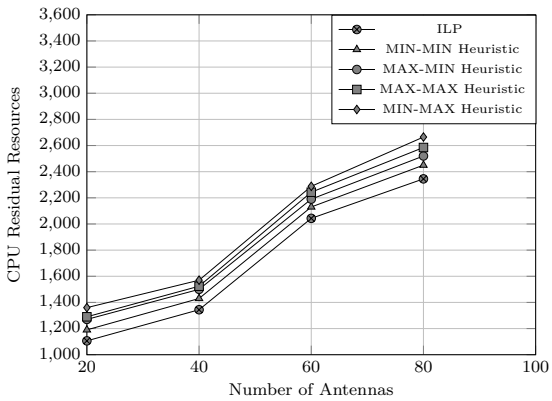


Figure 5.7: CPU residual resources behavior

Figure 5.8 illustrates the total and incurred latency when increasing the number of antennas. We observe that this latency, for the **ILP**, ***MAX-MAX*** and ***MIN-MAX*** solutions is in general close to zero and this is due to the placement of the BBU functions at the same physical node. In fact, we assume that if two or more BBU functions are placed in the same physical node, then the necessary latency between these functions is negligible (close to zero). Nevertheless, the solution provided by the *MIN-MIN* and *MAX-MIN* algorithms generate some latency (capped by 60μsec in the worst case) to guarantee the chaining of the deployed BBU functions.

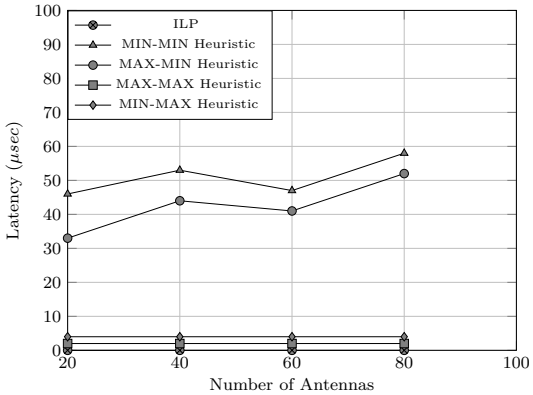


Figure 5.8: Latency behavior

The performance assessment would not be complete without addressing the scalability of our proposed algorithms for large problem instances. We illustrate in Table 5.3 the obtained simulation results regarding the scalability of the exact approach based on ILP formulation and the heuristic algorithms evaluated on Euclidean graphs.

Table 5.3: Scalability and convergence time comparison using Euclidean graphs

#Antennas	**#Edge Clouds**	**Heuristic Variant**	**Heuristic Time (s)**	**ILP Time**
100	**10**	MIN-MIN	2.25	**29.11min**
		MIN-MAX	2.49	
		MAX-MIN	4.35	
		MAX-MAX	2.87	
	15	MIN-MIN	2.98	**33.47min**
		MIN-MAX	3.26	
		MAX-MIN	6.17	
		MAX-MAX	4.00	
	20	MIN-MIN	4.19	**42.85min**
		MIN-MAX	3.98	
		MAX-MIN	7.99	
		MAX-MAX	4.86	
200	**10**	MIN-MIN	42.51	**>12h**
		MIN-MAX	No solution*	
		MAX-MIN	82.15	
		MAX-MAX	35.43	
	15	MIN-MIN	<1min	**>15h**
		MIN-MAX	No solution*	
		MAX-MIN	<2min	
		MAX-MAX	43.01s	
	20	MIN-MIN	<1min	**>16h**
		MIN-MAX	<1min	
		MAX-MIN	<2min	
		MAX-MAX	<1min	

* BBU functions chains cannot be all deployed.

Table 5.3 shows a significant gap between the convergence times of the four heuristic algorithms and the ILP solution. This is expected as the ILP explores all

feasible solutions. Clearly, our approximation algorithms can reach the optimal or near-optimal solutions with substantially less computation time.

In some cases as shown in Table 5.3, for example, the case of 200 antennas and 10 edge clouds, the *MIN-MAX* algorithm is not able to find a solution and this is due to two reasons: (i) there is not enough resources to host the demands and (ii) the *MIN-MAX* algorithm cannot explore certain feasible solutions. Moreover, we observe that the *MAX-MIN* algorithm requires slightly more computation time than those of the other heuristic algorithms. This is due to the fact that the *MAX-MIN* algorithm is deploying the different BBU functions on different servers necessitating the usage of a shortest path before reaching a solution meeting the latency requirements.

5.6 Conclusion

In this chapter, we studied the BBU function split placement problem represented as a mapping of BBU function chains on a hierarchical network topology modeled as a multi-stage graph. We proposed an exact formulation based on integer linear programming approach to describe the convex hull of the addressed problem. However, this optimization is known to not to scale for large problem instances due to the NP-Hardness of the BBU function split placement problem. For a large network size, we proposed new approximation algorithms based on the construction of an extended multi-stage graph to optimally deploy the requested chains in the network architecture when taking into account the high latency requirements and the limited processing capacity of physical nodes (in the centralized data centers).

The evaluation results revealed the efficiency of the *MAX-MAX* and *MIN-MAX* algorithms in terms of optimal solutions and convergence time even in the case of a large network size. The *MIN-MIN* and *MAX-MIN* algorithms are also favorable in terms of near-optimal solutions and convergence time.

The next chapter will be dedicated to summarize all our contributions of the previous chapters, and will highlight the important research challenges that we would like to address in the future.

Chapter 6

Conclusions and perspectives

In this chapter, we summarize our contributions for C-RAN optimization and propose some open future research topics.

Conclusions and main contributions

We addressed in this manuscript resource allocation problems in the context of C-RAN by proposing new scalable and cost-efficient algorithms based on combinatorial optimization techniques. In fact, C-RAN is considered as a promising network architecture for 5G mobile networks to meet diverse service requirements and to deal with new business opportunities, e.g. eMBB, mMTC, URLLC, etc. C-RAN is expected to reduce network costs, e.g. CAPEX and OPEX, and improve the resource utilization efficiency.

To achieve these goals, we investigated in Chapter 1 new algorithms to assign antennas (RRHs) demands to available edge data centers when latency and processing requirements are met. Our proposal aims to optimally address the RRH-BBU assignment problem by jointly optimizing the resource utilization and the communication latency on the fronthaul network. We modeled this problem using an exact ILP formulation to determine the most appropriate strategies in RRH-BBU assignment. This exact algorithm optimizes the resource consumption (in terms of active edge data centers) and communication latency associated for assigning antennas demands to available edge data centers. We proved also that the addressed problem is NP-Hard. Thus, in order to address large problem instances, we proposed three approximation algorithms : matroid-based approach, b-matching-based formulation and multiple knapsack-based algorithm to meet a larger number of antennas demands in negligible times. The performance evaluation revealed that the matroid and b-matching algorithms can rapidly find good RRH-BBU assignment solutions when achieving resource utilization gains.

However, such gains can be achieved only when reducing the inter-cell interference caused by the high density of cells in C-RAN. To consider these constraints, we proposed a complete mathematical formulation based on Branch-and-Cut methods to jointly minimize the levels of inter-cell interference and maintain the full network

coverage. We added new valid inequalities, i.e. chordless cycles and connectivity constraints, for our optimization model in order to reduce the space solution, i.e. convex hull, of the addressed problem and then accelerate the necessary convergence time to obtain optimum solutions. We evaluated the performance of our proposed approach by comparing with Rips-based approach which is considered as one of the most efficient algorithms to address the full network coverage problem. We considered different simulation scenarios and a real cellular network, e.g. small area in Paris, to evaluate the performance of our proposal. In both cases, the obtained results showed the efficiency of our approach that performs consistently well across all scenarios and performance metrics and proved its ability in providing good solutions that jointly optimize the full network coverage and minimize the inter-cell interference caused by network densification.

In addition to the increase of inter-cell interference levels, improving the existing density of cells by deploying more antennas will significantly increase the amount of baseband processing needed to meet the growing number of antennas demands. This leads also to increase the amount of data traffic demands, with strict latency and bandwidth constraints, between antennas and centralized data centers. To address these issues, we discussed in Chapter 5 how to find best trade-offs between benefits of C-RAN in terms of baseband processing centralization and strong latency requirements on fronthaul links, which represent the main obstacle for the deployment of C-RAN. For that, we investigated in Chapter 5 new algorithms to determine the optimal locations of BBU processing functions between cell sites (RRHs) and BBUs when considering 3GPP solution, outlined as the best split option proposal, to relax the latency and bandwidth requirements on the fronthaul network. We proposed an exact formulation based on ILP modeling and heuristic algorithms to provide optimal or near optimal solutions. Then, we highlighted the performance of each proposed algorithm in terms of optimal solutions, convergence time and scalability.

Future research directions

In the following, we propose some open research challenges that we would like to address in the future :

- For sake of simplicity, we only considered the communication latency on the fronthaul network joining antennas (RRHs) and centralized data centers (BBU pools) to model our resource allocation problems in the context of C-RAN. It would be very interesting to consider also the BBU processing time (compute latency) required to perform different BBU functions co-located in the edge data centers. This can lead to nonlinear objective functions that should be efficiently optimized. The problem becomes more complex and requires depth studies relying on Lagrangian relaxations, for instance. Furthermore, the data traffic on the fronthaul network, which connects the antennas and the centralized data centers, can be transmitted using different protocols including CPRI and OBSAI. The fronthaul network can be realized by different technologies, such as optical fiber communication, standard wireless communication,

or mmWave communication. The impact of these protocols and technologies can be investigated to better evaluate the performance of our proposed models and algorithms.

- In Chapter 4, we discussed how to jointly reduce inter-cell interference and guarantee the full network coverage for C-RAN. In order to achieve these conflicting goals, we modeled the coverage of each antenna by circular area with various coverage radii. However, in real life networks, antennas do not have regular shapes and their coverage area depends on geographic, environmental and network parameters. As future work, our problem modeling can be extended by taking into account the irregularity of antennas' coverage area to better evaluate the performance of our approach when considering real life constraints. To reach these objectives, new mathematical modeling of these constraints should be investigated.

- The performance of our proposed exact ILP formulation for full network coverage problem (Chapter 4) has been evaluated using different network topologies. Among these networks, we identified some network variants, which are only composed by cliques with at most four edges, that can be solved optimally in negligible times when the integrity constraints are relaxed. Hence, new mathematical formulation with valid inequalities can be investigated to characterize polynomial time variants of the NP-Complete full network coverage problem.

- Machine learning approaches can be used to address resource allocation problems in the context of C-RAN. In fact, a large amount of numerical results has been collected from our different simulation scenarios. These simulation results can be exploited using machine learning algorithms to improve the quality of the solutions founded by our proposed optimization algorithms. In fact, machine learning techniques have been recently used in many mathematical optimization to accelerate the necessary convergence time to find optimal solutions (see for instance [113], [114] and [115]). Hence, hybrid machine-learning and optimization methods can be used to improve the performance of our proposed algorithms for resource allocation problems in the context of C-RAN.

- Network slicing is another challenge that has the merit to be addressed to enable the deployment of next generation mobile networks (5G). In fact, network slicing is considered as one of the key enablers to enhance the flexibility of C-RAN and to meet new 5G services and opportunities. It consists in deploying multiple logical networks over a shared physical infrastructure, and then providing as a service or slice. In Chapter 5, exact and heuristic algorithms are proposed to address the problem of the optimal placement of BBU function chains on a shared network infrastructure. This problem is very similar to the problem of how to allocate the shared resources to slices. Similarly to our proposed algorithms, we can propose new optimization approaches to address the network slicing challenges in the context of C-RAN.

Bibliography

[1] "5G Security Innovation with Cisco." White Paper, 2018.

[2] C. Qiu and H. Shen, "A Delaunay-Based Coordinate-Free Mechanism for Full Coverage in Wireless Sensor Networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, pp. 828–839, April 2014.

[3] Paris 4G LTE Map. `https://www.anfr.fr/gestion-des-frequences-sites/lobservatoire-2g-3g-4g/lobservatoire-en-carte2/#menu2`, 2018.

[4] A. Fischer, J. F. Botero, M. T. Beck, H. de Meer, and X. Hesselbach, "Virtual Network Embedding: A Survey," *IEEE Communications Surveys Tutorials*, vol. 15, pp. 1888–1906, Fourth 2013.

[5] 3GPP, "Study on Scenarios and Requirements for Next Generation Access Technologies," TR 38.913 version 14.2.0 Release 14, 3GPP, May 2017.

[6] G. Kardaras and C. Lanzani, "Advanced multimode radio for wireless mobile broadband communication," in *2009 European Wireless Technology Conference*, pp. 132–135, Sep. 2009.

[7] N. Nikaein, "Processing Radio Access Network Functions in the Cloud: Critical Issues and Modeling," in *Proceedings of the 6th International Workshop on Mobile Cloud Computing and Services*, pp. 36–43, 2015.

[8] "Open Base Station Architecture Initiative." BTS System Reference Document Version 2.0, 2006.

[9] A. de la Oliva, J. A. Hernandez, D. Larrabeiti, and A. Azcorra, "An overview of the CPRI specification and its application to C-RAN-based LTE scenarios," *IEEE Communications Magazine*, vol. 54, pp. 152–159, February 2016.

[10] T. S. Rappaport, S. Sun, R. Mayzus, H. Zhao, Y. Azar, K. Wang, G. N. Wong, J. K. Schulz, M. Samimi, and F. Gutierrez, "Millimeter Wave Mobile Communications for 5G Cellular: It Will Work!," *IEEE Access*, vol. 1, pp. 335–349, 2013.

[11] K. Chen and R. Duan, "C-RAN : The Road Towards Green RAN," Tech. Rep. V3.0, December 2013.

[12] J. Wu, Z. Zhang, Y. Hong, and Y. Wen, "Cloud radio access network (C-RAN): a primer," *IEEE Network*, vol. 29, pp. 35–41, Jan 2015.

[13] M. Peng, Y. Sun, X. Li, Z. Mao, and C. Wang, "Recent Advances in Cloud Radio Access Networks: System Architectures, Key Techniques, and Open Issues," *IEEE Communications Surveys Tutorials*, vol. 18, pp. 2282–2308, thirdquarter 2016.

[14] "The Benefits of Cloud-RAN Architecture in Mobile Network Expansion." White Paper, 2014.

[15] S. Brueck, L. Zhao, J. Giese, and M. A. Amin, "Centralized scheduling for joint transmission coordinated multi-point in LTE-Advanced," in *2010 International ITG Workshop on Smart Antennas (WSA)*, pp. 177–184, Feb 2010.

[16] G. Boudreau, J. Panicker, N. Guo, R. Chang, N. Wang, and S. Vrzic, "Interference coordination and cancellation for 4G networks," *IEEE Communications Magazine*, vol. 47, pp. 74–81, April 2009.

[17] NGMN, "RAN evolution project backhaul and fronthaul evolution," *NGMN Alliance*, 2015.

[18] R. Mijumbi, J. Serrat, J. Gorricho, N. Bouten, F. De Turck, and R. Boutaba, "Network Function Virtualization: State-of-the-Art and Research Challenges," *IEEE Communications Surveys Tutorials*, vol. 18, pp. 236–262, Firstquarter 2016.

[19] X. Costa-Perez, J. Swetina, T. Guo, R. Mahindra, and S. Rangarajan, "Radio access network virtualization for future mobile carrier networks," *IEEE Communications Magazine*, vol. 51, pp. 27–35, July 2013.

[20] N. Nikaein, M. Marina, S. Manickam, A. Dawson, R. Knopp, and C. Bonnet, "OpenAirInterface: A flexible platform for 5G research," *ACM Sigcomm Computer Communication Review, Volume 44, NÂ°5, October 2014*, 10 2014.

[21] EURECOM, Open Air Interface. `http://www.openairinterface.org/`, 2014.

[22] X. Wei, H. Liu, Z. Geng, K. Zheng, R. Xu, Y. Liu, and P. Chen, "Software Defined Radio Implementation of a Non-Orthogonal Multiple Access System Towards 5G," *IEEE Access*, vol. 4, pp. 9604–9613, 2016.

[23] S. Sun, M. Kadoch, L. Gong, and B. Rong, "Integrating network function virtualization with SDR and SDN for 4G/5G networks," *IEEE Network*, vol. 29, pp. 54–59, May 2015.

[24] Z. Feng, C. Qiu, Z. Feng, Z. Wei, W. Li, and P. Zhang, "An effective approach to 5G: Wireless network virtualization," *IEEE Communications Magazine*, vol. 53, pp. 53–59, Dec 2015.

[25] J. G. Oxley, *Matroid Theory (Oxford Graduate Texts in Mathematics)*. New York, NY, USA: Oxford University Press, Inc., 2006.

[26] B. Korte and J. Vygen, *b-Matchings and T-Joins*, pp. 305–324. Springer Publishing Company, Incorporated, 6th ed., 2018.

[27] A. Li, Y. Sun, X. Xu, and C. Yuan, "An energy-effective network deployment scheme for 5G Cloud Radio Access Networks," in *2016 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, pp. 684–689, April 2016.

[28] X. Xu, J. Liu, W. Chen, Y. Hou, and X. Tao, "Storage and computing resource enabled joint virtual resource allocation with QoS guarantee in mobile networks," *Science China Information Sciences*, vol. 60, p. 040304, Mar 2017.

[29] L. Pu, L. Jiao, X. Chen, L. Wang, Q. Xie, and J. Xu, "Online Resource Allocation, Content Placement and Request Routing for Cost-Efficient Edge Caching in Cloud Radio Access Networks," *IEEE Journal on Selected Areas in Communications*, vol. 36, pp. 1751–1767, Aug 2018.

[30] H. Kellerer, U. Pferschy, and D. Pisinger, *Knapsack Problems*. Springer Publishing Company, 2004.

[31] N. Mharsi and M. Hadji, "Joint Optimization of Communication Latency and Resource Allocation in Cloud Radio Access Networks," in *2018 International Conference on Smart Communications in Network Technologies (SaCoNeT)*, pp. 13–18, Oct 2018.

[32] N. Mharsi and M. Hadji, "Edge computing optimization for efficient RRH-BBU assignment in Cloud Radio Access Networks," *Computer Networks*, 2019.

[33] R. Ghrist and A. Muhammad, "Coverage and hole-detection in sensor networks via homology," in *Fourth International Symposium on Information Processing in Sensor Networks*, pp. 254–260, April 2005.

[34] N. Mharsi, M. Hadji, and P. Martins, "Full Coverage Hole Optimization in Cloud Radio Access Networks," in *IEEE Global Communications Conference, GLOBECOM 2018, Abu Dhabi, United Arab Emirates, December 9-13, 2018*, pp. 1–7, 2018.

[35] N. Mharsi and M. Hadji, "A mathematical programming approach for full coverage hole optimization in Cloud Radio Access Networks," *Computer Networks*, vol. 150, pp. 117–126, 2019.

[36] N. Mharsi, M. Hadji, D. Niyato, W. Diego, and R. Krishnaswamy, "Scalable and cost-efficient algorithms for baseband unit (BBU) function split placement," in *2018 IEEE Wireless Communications and Networking Conference, WCNC 2018, Barcelona, Spain, April 15-18, 2018*, pp. 1–6, 2018.

[37] H. Marchand, A. Martin, R. Weismantel, and L. Wolsey, "Cutting planes in integer and mixed integer programming," *Discrete Applied Mathematics*, vol. 123, no. 1, pp. 397 – 446, 2002.

[38] R. Wilson, *Introduction to Graph Theory.* Longman, 1996.

[39] B. Korte and J. Vygen, *Combinatorial Optimization: Theory and Algorithms.* Springer Publishing Company, Incorporated, 6th ed., 2018.

[40] E. Lawler, *Combinatorial Optimization: Networks and Matroids.* Dover Books on Mathematics, Dover Publications, 2012.

[41] L. Matthews, "*Bicircular matroids*," *Quart. J. Math. Oxford.*, vol. 28, pp. 213–228, 1977.

[42] R. J. Wilson, "An introduction to matroid theory," *The American Mathematical Monthly*, vol. 80, no. 5, pp. 500–525, 1973.

[43] M. W. Padberg and M. R. Rao, "Odd Minimum Cut-Sets and b-Matchings," *Mathematics of Operations Research*, vol. 7, no. 1, pp. 67–80, 1982.

[44] A. Letchford, G. Reinelt, and D. Theis, "Odd Minimum Cut Sets and b-Matchings Revisited," *SIAM Journal on Discrete Mathematics*, vol. 22, no. 4, pp. 1480–1487, 2008.

[45] M. R. Garey and D. S. Johnson, *Computers and Intractability; A Guide to the Theory of NP-Completeness.* New York, NY, USA: W. H. Freeman & Co., 1990.

[46] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms, Third Edition.* The MIT Press, 3rd ed., 2009.

[47] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *The Bellman Ford algorithm*, pp. 588–592. The MIT Press, 3rd ed., 2009.

[48] B. V. Cherkassky, A. V. Goldberg, and T. Radzik, "Shortest paths algorithms: Theory and experimental evaluation," *Mathematical Programming*, vol. 73, pp. 129–174, May 1996.

[49] L. M. Goldschlager, R. A. Shaw, and J. Staples, "The maximum flow problem is log space complete for p," *Theoretical Computer Science*, vol. 21, no. 1, pp. 105 – 111, 1982.

[50] R. M. Karp, *Reducibility among Combinatorial Problems*, pp. 85–103. Boston, MA: Springer US, 1972.

[51] B. Korte and J. Vygen, *The Knapsack Problem*, pp. 471–488. Springer Publishing Company, Incorporated, 6th ed., 2018.

[52] M. A. Trick, "A dynamic programming approach for consistency and propagation for knapsack constraints," *Annals of Operations Research*, vol. 118, pp. 73–84, Feb 2003.

[53] A. Hatcher, *Algebraic topology*. Cambridge: Cambridge University Press, 2002.

[54] F. Yan, P. Martins, and L. Decreusefond, "Accuracy of homology based approaches for coverage hole detection in wireless sensor networks," in *2012 IEEE International Conference on Communications (ICC)*, pp. 497–502, June 2012.

[55] M. Agiwal, A. Roy, and N. Saxena, "Next Generation 5G Wireless Networks: A Comprehensive Survey," *IEEE Communications Surveys Tutorials*, vol. 18, pp. 1617–1655, thirdquarter 2016.

[56] R. Mijumbi, J. Serrat, J. Gorricho, J. Rubio-Loyola, and S. Davy, "Server placement and assignment in virtualized radio access networks," in *2015 11th International Conference on Network and Service Management (CNSM)*, pp. 398–401, Nov 2015.

[57] N. Yu, Z. Song, H. Du, H. Huang, and X. Jia, "Multi-resource allocation in cloud radio access networks," in *2017 IEEE International Conference on Communications (ICC)*, pp. 1–6, May 2017.

[58] D. Mishra, P. C. Amogh, A. Ramamurthy, A. A. Franklin, and B. R. Tamma, "Load-aware dynamic RRH assignment in Cloud Radio Access Networks," in *2016 IEEE Wireless Communications and Networking Conference*, April 2016.

[59] K. Wang, W. Zhou, and S. Mao, "On Joint BBU/RRH Resource Allocation in Heterogeneous Cloud-RANs," *IEEE Internet of Things Journal*, vol. 4, pp. 749–759, June 2017.

[60] E. Aqeeli, A. Moubayed, and A. Shami, "Power-Aware Optimized RRH to BBU Allocation in C-RAN," *IEEE Transactions on Wireless Communications*, vol. 17, pp. 1311–1322, Feb 2018.

[61] H. Holm, A. Checko, R. Al-obaidi, and H. Christiansen, "Optimal assignment of cells in C-RAN deployments with multiple BBU pools," in *2015 European Conference on Networks and Communications (EuCNC)*, pp. 205–209, June 2015.

[62] K. Boulos, M. E. Helou, K. Khawam, M. Ibrahim, S. Martin, and H. Sawaya, "RRH clustering in cloud radio access networks with re-association consideration," in *2018 IEEE Wireless Communications and Networking Conference (WCNC)*, pp. 1–6, April 2018.

[63] J. Yao and N. Ansari, "QoS-Aware Joint BBU-RRH Mapping and User Association in Cloud-RANs," *IEEE Transactions on Green Communications and Networking*, vol. 2, pp. 881–889, Dec 2018.

[64] E. G. Coffman, M. R. Garey, and D. S. Johnson, *Approximation Algorithms for Bin-Packing — An Updated Survey*, pp. 49–106. Vienna: Springer Vienna, 1984.

[65] J. Tang, W. P. Tay, and T. Q. S. Quek, "Cross-Layer Resource Allocation With Elastic Service Scaling in Cloud Radio Access Network," *IEEE Transactions on Wireless Communications*, vol. 14, pp. 5068–5081, Sept 2015.

[66] M. Khan, R. S. Alhumaima, and H. S. Al-Raweshidy, "Reducing energy consumption by dynamic resource allocation in C-RAN," in *2015 European Conference on Networks and Communications (EuCNC)*, pp. 169–174, June 2015.

[67] Y. Zhong, T. Q. S. Quek, and W. Zhang, "Complementary Networking for C-RAN: Spectrum Efficiency, Delay and System Cost," *IEEE Transactions on Wireless Communications*, vol. 16, pp. 4639–4653, July 2017.

[68] N. Bhushan, J. Li, D. Malladi, R. Gilmore, D. Brenner, A. Damnjanovic, R. T. Sukhavasi, C. Patel, and S. Geirhofer, "Network densification: the dominant theme for wireless evolution into 5G," *IEEE Communications Magazine*, vol. 52, pp. 82–89, February 2014.

[69] X. Yu, M. Xu, L. Cheng, and N. Hu, "A novel coverage holes detection and holes recovery algorithm in wireless sensor networks," in *The 27th Chinese Control and Decision Conference (2015 CCDC)*, pp. 3640–3644, May 2015.

[70] Y. Tian, X. Wang, Y. Jiang, and G. You, "A distributed probabilistic coverage sets configuration method for high density WSN," in *Chinese Automation Congress (CAC)*, pp. 2312–2316, Oct 2017.

[71] A. Daoudi, B. Detienne, R. E. Azouzi, I. Benelallam, and E. H. Bouyakhf, "Robust coverage optimization approach in Wireless Sensor Networks," in *International Conference on Wireless Networks and Mobile Communications (WINCOM)*, pp. 1–7, Nov 2017.

[72] A. Vergne, L. Decreusefond, and P. Martins, "Reduction algorithm for simplicial complexes," in *Proceedings IEEE INFOCOM*, pp. 475–479, April 2013.

[73] V. de Silva and R. Ghrist, "Coordinate-free Coverage in Sensor Networks with Controlled Boundaries via Homology," *The International Journal of Robotics Research*, vol. 25, no. 12, pp. 1205–1222, 2006.

[74] N. Le, P. Martins, L. Decreusefond, and A. Vergne, "Simplicial homology based energy saving algorithms for wireless networks," in *IEEE International Conference on Communication Workshop (ICCW)*, pp. 166–172, June 2015.

[75] H. Yang, J. Zhang, Y. Ji, and Y. Lee, "C-RoFN: multi-stratum resources optimization for cloud-based radio over optical fiber networks," *IEEE Communications Magazine*, vol. 54, pp. 118–125, August 2016.

[76] H. Yang, J. Zhang, Y. Ji, Y. He, and Y. Lee, "Experimental demonstration of multi-dimensional resources integration for service provisioning in cloud radio over fiber network.," *Scientific Reports*, vol. 6, 2016.

[77] NGMN, "Project RAN Evolution: Further Study on Critical C-RAN Technologies," Technical Document v1, NGMN Alliance, March 2015.

[78] R3-162854, "RAN functional split considerations and preferences," RAN WG3 Meeting #94, 3GPP, November 2016.

[79] C. I, J. Huang, R. Duan, C. Cui, J. . Jiang, and L. Li, "Recent Progress on C-RAN Centralization and Cloudification," *IEEE Access*, vol. 2, pp. 1030–1039, 2014.

[80] A. Checko, A. P. Avramova, M. S. Berger, and H. L. Christiansen, "Evaluating C-RAN fronthaul functional splits in terms of network level energy and cost savings," *Journal of Communications and Networks*, vol. 18, pp. 162–172, April 2016.

[81] X. Wang, L. Wang, S. E. Elayoubi, A. Conte, B. Mukherjee, and C. Cavdar, "Centralize or distribute? A techno-economic study to design a low-cost cloud radio access network," in *2017 IEEE International Conference on Communications (ICC)*, pp. 1–7, May 2017.

[82] H. Holm, A. Checko, R. Al-obaidi, and H. Christiansen, "Optimal assignment of cells in C-RAN deployments with multiple BBU pools," in *2015 European Conference on Networks and Communications (EuCNC)*, pp. 205–209, June 2015.

[83] J. Liu, S. Zhou, J. Gong, Z. Niu, and S. Xu, "Graph-based Framework for Flexible Baseband Function Splitting and Placement in C-RAN," in *IEEE ICC 2015 - Wireless Communications Symposium*, 2015.

[84] N. Nikaein, "Processing Radio Access Network Functions in the Cloud: Critical Issues and Modeling," in *Proceedings of the 6th International Workshop on Mobile Cloud Computing and Services*, MCS '15, (New York, NY, USA), pp. 36–43, ACM, 2015.

[85] A. Checko, H. L. Christiansen, Y. Yan, L. Scolari, G. Kardaras, M. S. Berger, and L. Dittmann, "Cloud RAN for Mobile Networks - A Technology Overview," *IEEE Communications Surveys Tutorials*, vol. 17, pp. 405–426, Firstquarter 2015.

[86] S. Bhaumik, S. P. Chandrabose, M. K. Jataprolu, G. Kumar, A. Muralidhar, P. A. Polakos, V. Srinivasan, and T. Woo, "CloudIQ: a framework for processing base stations in a data center," in *MobiCom*, 2012.

[87] D. G. Cattrysse and L. N. V. Wassenhove, "A survey of algorithms for the generalized assignment problem," *European Journal of Operational Research*, vol. 60, no. 3, pp. 260 – 272, 1992.

[88] S. Martello and P. Toth, *Knapsack Problems: Algorithms and Computer Implementations*. John Wiley and Sons 1 edition, 1990.

[89] R. E. Korf, "A New Algorithm for Optimal Bin Packing," in *Eighteenth National Conference on Artificial Intelligence*, (Menlo Park, CA, USA), pp. 731–736, American Association for Artificial Intelligence, 2002.

[90] M. L. Fisher, R. Jaikumar, and L. N. V. Wassenhove, "A multiplier adjustment method for the generalized assignment problem," *Management Science*, vol. 32, no. 9, pp. 1095–1103, 1986.

[91] F. Musumeci, C. Bellanzon, N. Carapellese, M. Tornatore, A. Pattavina, and S. Gosselin, "Optimal BBU Placement for 5G C-RAN Deployment Over WDM Aggregation Networks," *Journal of Lightwave Technology*, vol. 34, pp. 1963–1970, April 2016.

[92] Y. Zhang and C. Leung, "Resource allocation in an OFDM-based cognitive radio system," *IEEE Transactions on Communications*, vol. 57, pp. 1928–1931, July 2009.

[93] S. Chuah, Z. Chen, and Y. Tan, "Energy-efficient resource allocation and scheduling for multicast of scalable video over wireless networks," *IEEE Transactions on Multimedia*, vol. 14, pp. 1324–1336, Aug 2012.

[94] IBM Cplex Optimizer. `https://www.ibm.com/analytics/data-science/prescriptive-analytics/cplex-optimizer`, 2018.

[95] Y. Li, H. Xia, S. Wu, and C. Lu, "Joint optimization of computing and radio resource under outage QoS constraint in C-RAN," in *2017 International Symposium on Wireless Communication Systems (ISWCS)*, pp. 107–111, Aug 2017.

[96] K. B. Baltzis and J. N. Sahalos, "On the statistical description of the AoA of the uplink interfering signals in a cellular communication system," *European Transactions on Telecommunications*, vol. 21, no. 2, pp. 187–194, 2010.

[97] X. Yang and A. O. Fapojuwo, "Performance analysis of hexagonal cellular networks in fading channels," *Wireless Communications and Mobile Computing*, vol. 16, no. 7, pp. 850–867, 2016.

[98] M. Maqbool, P. Godlewski, M. Coupechoux, and J.-M. Kélif, "Analytical Performance Evaluation of Various Frequency Reuse and Scheduling Schemes in Cellular OFDMA Networks," *Perform. Eval.*, vol. 67, pp. 318–337, Apr. 2010.

[99] X. Y. Li, P. J. Wan, and O. Frieder, "Coverage in Wireless Ad Hoc Sensor Networks," *IEEE Trans. Comput.*, vol. 52, pp. 753–763, June 2003.

[100] R. W. Heath, M. Kountouris, and T. Bai, "Modeling Heterogeneous Network Interference Using Poisson Point Processes," *IEEE Transactions on Signal Processing*, vol. 61, pp. 4114–4126, Aug 2013.

[101] M. Taranetz and M. Rupp, "A Circular Interference Model for Heterogeneous Cellular Networks," *IEEE Transactions on Wireless Communications*, vol. 15, pp. 1432–1444, Feb 2016.

[102] E. Tuba, M. Tuba, and M. Beko, "Mobile wireless sensor networks coverage maximization by firefly algorithm," in *International Conference Radioelektronika (RADIOELEKTRONIKA)*, pp. 1–5, April 2017.

[103] N. Phan, T. Bui, H. Jiang, P. Li, Z. Pan, and N. Liu, "Coverage optimization of LTE networks based on antenna tilt adjusting considering network load," *China Communications*, vol. 14, pp. 48–58, May 2017.

[104] W. Mulzer and G. Rote, "Minimum-weight Triangulation is NP-hard," *J. ACM*, vol. 55, pp. 11:1–11:29, May 2008.

[105] S. D. Nikolopoulos and L. Palios, "Detecting Holes and Antiholes in Graphs," *Algorithmica*, vol. 47, pp. 119–138, Feb 2007.

[106] N. Sokhn, R. Baltensperger, L.-F. Bersier, J. Hennebert, and U. Ultes-Nitsche, "Identification of chordless cycles in ecological networks," in *Complex Sciences* (K. Glass, R. Colbaugh, P. Ormerod, and J. Tsao, eds.), (Cham), pp. 316–324, Springer International Publishing, 2013.

[107] D. R. Ford and D. R. Fulkerson, *Flows in Networks.* Princeton, NJ, USA: Princeton University Press, 2010.

[108] 3GPP, "Study on New Radio Access Technology; Radio Access Architecture and Interfaces," TR 38.801 v2.0.0 Release 14, 3GPP, March 2017.

[109] F. Moufida, B. Guyader, P. Varga, A. Gravey, S. Gosselin, and J. Torrijos Gijon, "Multi-Criteria Comparison Between Legacy and Next Generation Point of Presence Broadband Network Architectures," *Advances in Computer Science: an International Journal*, vol. Vol.4, pp. 126–140, 05 2015.

[110] E. Amaldi, S. Coniglio, A. M. Koster, and M. Tieves, "On the computational complexity of the virtual network embedding problem," *Electronic Notes in Discrete Mathematics*, vol. 52, pp. 213 – 220, 2016.

[111] Y. Zhu and M. Ammar, "Algorithms for Assigning Substrate Network Resources to Virtual Network Components," in *Proceedings IEEE INFOCOM 2006. 25TH IEEE International Conference on Computer Communications*, pp. 1–12, April 2006.

[112] M. Yu, Y. Yi, J. Rexford, and M. Chiang, "Rethinking Virtual Network Embedding: Substrate Support for Path Splitting and Migration," *SIGCOMM Comput. Commun. Rev.*, vol. 38, pp. 17–29, Mar. 2008.

[113] E. B. Khalil, P. L. Bodic, L. Song, G. L. Nemhauser, and B. N. Dilkina, "Learning to branch in mixed integer programming," in *AAAI*, 2016.

[114] A. M. Alvarez, Q. Louveaux, and L. Wehenkel, "A machine learning-based approximation of strong branching," *INFORMS J. on Computing*, vol. 29, pp. 185–195, Feb. 2017.

[115] Y. Zhou, J.-K. Hao, and B. Duval, "Reinforcement learning based local search for grouping problems: A case study on graph coloring," *Expert Systems with Applications*, vol. 64, pp. 412 – 422, 2016.

ÉCOLE DOCTORALE

Sciences et technologies de l'information et de la communication (STIC)

Titre : Cloud-Radio Access Networks : Conception, Optimisation et Algorithmes

Mots clés : C-RAN, 5G, Optimisation Combinatoire, Allocation des Ressources

Résumé : Cloud-Radio Access Network (C-RAN) est une architecture prometteuse pour faire face à l'augmentation exponentielle des demandes de trafic de données et surmonter les défis des réseaux de prochaine génération (5G). Le principe de base de C-RAN consiste à diviser la station de base traditionnelle en deux entités : les unités de bande de base (BaseBand Unit, BBU) et les têtes radio distantes (Remote Radio Head, RRH) et à mettre en commun les BBUs de plusieurs stations dans des centres de données centralisés (pools de BBU). Ceci permet la réduction des coûts d'exploitation, l'amélioration de la capacité du réseau ainsi que des gains en termes d'utilisation des ressources. Pour atteindre ces objectifs, les opérateurs réseaux ont besoin d'investiguer de nouveaux algorithmes pour les problémes d'allocation de ressources permettant ainsi de faciliter le déploiement de l'architecture C-RAN. La plupart de ces problèmes sont très complexes et donc très difficiles à résoudre. Par conséquent, nous utilisons l'optimisation combinatoire qui propose des outils puissants pour adresser ce type des problèmes.
Un des principaux enjeux pour permettre le déploiement du C-RAN est de déterminer une affectation optimale des RRHs (antennes) aux centres de données centralisés (BBUs) en optimisant conjointement la latence sur le réseau de transmission fronthaul et la consommation des ressources. Nous modélisons ce problème à l'aide d'une formulation mathématique basée sur une approche de programmation linéaire en nombres entiers permettant de determiner les stratégies optimales pour le problème d'affectation des ressources entre RRH-BBU et nous proposons également des heuristiques afin de pallier la difficuté au sens de la complexité algorithmique quand des instances larges du probléme sont traitées, permettant ainsi le passage à l'échelle. Une affectation optimale des antennes aux BBUs réduit la latence de communication attendue et offre des gains en termes d'utilisation des ressources. Néanmoins, ces gains dépendent fortement de l'augmentation des niveaux d'interférence inter-cellulaire causés par la densité élevée des antennes déployées dans les rèseaux C-RANs. Ainsi, nous proposons une formulation mathématique exacte basée sur les méthodes Branch-and-Cut qui consiste à consolider et ré-optimiser les rayons de couverture des antennes afin de minimiser les interférences inter-cellulaires et de garantir une couverture maximale du réseau conjointement. En plus de l'augmentation des niveaux d'interférence, la densité élevée des cellules dans le réseau C-RAN augmente le nombre des fonctions BBUs ainsi que le trafic de données entre les antennes et les centres de données centralisés avec de fortes exigences en terme de latence sur le réseau fronthaul. Par conséquent, nous discutons dans la troisième partie de cette thèse comment placer d'une maniére optimale les fonctions BBUs en considérant la solution split du 3GPP afin de trouver le meilleur compromis entre les avantages de la centralisation dans C-RAN et les forts besoins en latence et bande passante sur le réseau fronthaul. Nous proposons des algorithmes (exacts et heuristiques) issus de l'optimisation combinatoire afin de trouver rapidement des solutions optimales ou proches de l'optimum, même pour des instances larges du problème.

Title : Cloud-Radio Access Networks : Design, Optimization and Algorithms

Keywords : C-RAN, 5G, Combinatorial Optimization, Resource Allocation

Abstract : Cloud Radio Access Network (C-RAN) has been proposed as a promising architecture to meet the exponential growth in data traffic demands and to overcome the challenges of next generation mobile networks (5G). The main concept of C-RAN is to decouple the BaseBand Units (BBU) and the Remote Radio Heads (RRH), and place the BBUs in common edge data centers (BBU pools) for centralized processing. This gives a number of benefits in terms of cost savings, network capacity improvement and resource utilization gains. However, network operators need to investigate scalable and cost-efficient algorithms for resource allocation problems to enable and facilitate the deployment of C-RAN architecture. Most of these problems are very complex and thus very hard to solve. Hence, we use combinatorial optimization which provides powerful tools to efficiently address these problems.
One of the key issues in the deployment of C-RAN is finding the optimal assignment of RRHs (or antennas) to edge data centers (BBUs) when jointly optimizing the fronthaul latency and resource consumption. We model this problem by a mathematical formulation based on an Integer Linear Programming (ILP) approach to provide the optimal strategies for the RRH-BBU assignment problem and we propose also low-complexity heuristic algorithms to rapidly reach good solutions for large problem instances. The optimal RRH-BBU assignment reduces the expected latency and offers resource utilization gains. Such gains can only be achieved when reducing the inter-cell interference caused by the dense deployment of cell sites. We propose an exact mathematical formulation based on Branch-and-Cut methods that enables to consolidate and re-optimize the antennas radii in order to jointly minimize inter-cell interference and guarantee a full network coverage in C-RAN. In addition to the increase of inter-cell interference, the high density of cells in C-RAN increases the amount of baseband processing as well as the amount of data traffic demands between antennas and centralized data centers when strong latency requirements on fronthaul network should be met. Therefore, we discuss in the third part of this thesis how to determine the optimal placement of BBU functions when considering 3GPP split option to find optimal tradeoffs between benefits of centralization in C-RAN and transport requirements. We propose exact and heuristic algorithms based on combinatorial optimization techniques to rapidly provide optimal or near-optimal solutions even for large network sizes.

Université Paris-Saclay
Espace Technologique / Immeuble Discovery
Route de l'Orme aux Merisiers RD 128 / 91190 Saint-Aubin, France